1983

A computer program for solving the parabolic equation using an Implicit Finite-Difference solution method and incorporating exact interface conditions.

Jaeger, Larry Ernest.

http://hdl.handle.net/10945/19687

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A COMPUTER PROGRAM FOR SOLVING THE PARABOLIC EQUATION
USING AN IMPLICIT FINITE-DIFFERENCE SOLUTION METHOD
INCORPORATING EXACT INTERFACE CONDITIONS

by

Larry Ernest Jaeger

September 1983

Thesis Advisor: A. B. Coppens

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Computer Program for Solving the Parabolic Equation Using an Implicit Finite-Difference Solution Method and Incorporating Exact Interface Conditions.		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September, 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Larry Ernest Jaeger		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE September, 1983
		13. NUMBER OF PAGES 112
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release, Distributed Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Parabolic Wave Equation, Underwater Acoustics, Implicit Finite-Difference, Shallow Water, ASW Shallow, Propagation Loss, Acoustic Propagation, Bottom Interaction

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An Implicit Finite-Difference (IFD) program that incorporates exact interface conditions has been developed for solving the parabolic equation. The model preserves continuity of pressure and continuity of the normal component of particle velocity at the interface between media having different sound speeds and densities. Interface conditions are preserved for horizontal and sloping interfaces along a user-specified bottom profile. Test cases are included to demonstrate the use of the model.

Approved for public release, distribution unlimited

A Computer Program for Solving the Parabolic Equation Using an Implicit Finite-Difference Solution Method and Incorporating Exact Interface Conditions

by

Larry Ernest Jaeger
Lieutenant, United States Coast Guard
B.S., University of Wisconsin-Madison, 1973
M.S., University of Wisconsin-Madison, 1974

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN PHYSICS

from the

NAVAL POSTGRADUATE SCHOOL
September, 1983

ABSTRACT

An Implicit Finite-Difference (IFD) program that incorporates exact interface conditions has been developed for solving the parabolic equation. The model preserves continuity of pressure and continuity of the normal component of particle velocity at the interface between media having different sound speeds and densities. Interface conditions are preserved for horizontal and sloping interfaces along a user-specified bottom profile. Test cases are included to demonstrate the use of the model.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Since its introduction to the underwater acoustics community (Hardin and Tappert, 1973), the parabolic wave equation has stimulated a considerable amount of interest. The first solution programs used a split-step fast Fourier transform method to solve the parabolic equation; however, other solution techniques have been developed (McDaniel and Lee, 1982). One of the motives for developing alternative solution techniques is that problems arise when the Fourier transform encounters an interface between two media having different sound speeds or densities (Lee and Botseas, 1982).

One alternative solution technique uses an implicit finite-difference (IFD) solution method. The IFD method is unconditionally stable and has the capability to incorporate desired interface conditions. Implicit finite-difference methods for solving parabolic equations have been studied extensively by many authors.

A computer program that utilizes the IFD method to solve the parabolic equation has been developed and is examined in detail in this thesis. The computer program predicts acoustic propagation loss in environments having user-specified bottom profiles. The program preserves continuity of pressure and continuity of the normal component of

particle velocity at an interface between media having different sound speeds and densities.

The program utilizes concepts developed by earlier authors. The use of the IFD method to solve the parabolic equation in underwater acoustics was developed by Lee and Papadakis (1979). The mathematical treatment of horizontal and sloping interfaces was developed by McDaniel and Lee (1982) and Lee and McDaniel (1983). And finally, the program utilizes some design features of an earlier computer program developed by Lee and Botseas (1982).

II. PARABOLIC EQUATION

A. INTRODUCTION

The parabolic equation is an approximation to the elliptical wave equation. The derivation of the parabolic equation begins with the reduced wave equation (Helmholtz equation) in the form

$$\nabla^2 p + k^2 = 0 \qquad 2.1$$

or

$$\nabla^2 p + k_o{}^2 n^2 p = 0 \qquad 2.2$$

where

- k = wave number (= w/c)
- k_o = reference wave number (= w/c_o)
- n = index of refraction (= c_o/c)
- p = time independent factor of complex pressure
- c = sound speed
- c_o = reference sound speed
- w = angular source frequency (= $2\pi f$)

For the case of cylindrical symmetry (2.2) becomes

$$p_{rr} + (1/r)\, p_r + p_{zz} + k_o{}^2 n^2 p = 0 \qquad 2.3$$

It is then assumed that p is of the form

$$p = u(r,z)\ S(r)$$

where u is a function of both range and depth and S is a function of range only. Substitution of (2.4) into (2.3) and separation of variables shows that S(r) must satisfy

Bessel's equation of zero-order. For exp (-iwt) time dependence and outgoing waves, the solution is the zeroth-order Hankel function of the first kind,

$$S(r) = H_o^{(1)}(k_o r).$$

Further, u(r,z) must satisfy

$$u_{rr} + u_{zz} + \left(\frac{1}{r} + \frac{2}{S} S_r\right) u_r + k_o(n^2 - 1) u = 0. \qquad 2.5$$

With the help of the far-field asymptotic approximation for the Hankel function, (2.5) can be reduced to

$$u_{rr} + u_{zz} + 2ik_o u_r + k_o^2(n^2 - 1) u = 0. \qquad 2.6$$

We now assume that u varies slowly with respect to range,

$$\left|u_{rr}\right| \ll \left|2 k_o u_r\right| . \qquad 2.7$$

combining (2.6) and (2.7) results in

$$u_{zz} + 2ik_o u_r + k_o^2(n^2 - 1) u = 0. \qquad 2.8$$

Rearranging (2.8) results in the parabolic equation in the form

$$u_r = a(k_o,r,z) u + b(k_o,r,z) u_{zz} \qquad 2.9$$

where

$$a(k_o,r,z) = (ik_o/2) [n^2(r,z) - 1]$$

$$b(k_o,r,z) = i/2k_o.$$

The assumption (2.7), fundamental to the parabolic equation method, is equivalent to neglecting back-scatter.

B. SPLIT-STEP FAST FOURIER TRANSFORM SOLUTION

1. Description

For the first few years after its introduction into the acoustical community the parabolic equation was solved

exclusively with the help of the split-step fast Fourier transform (SSFFT) method developed by Tappert and Hardin (Jensen and Krol, 1975). In this method, u_{zz} in (2.8) is represented by the inverse transform of its Fourier transform. The SSFFT method requires periodic boundary conditions in z because of the finite Fourier transform. This is handled by introducing an artificial, horizontal, pressure release bottom below the physical bottom. The SSFFT method is unconditionally stable (Brock, 1978).

The SSFFT method has been implemented by Jensen and Krol and by Brock. Detailed descriptions can be found in publications of Jensen and Krol (1975) and Brock (1978).

2. Interface Treatment

Errors introduced by the SSFFT method are proportional to the range step and to n_{zz} where n is the index of refraction (Jensen and Krol, 1975). Because the index of refraction has a large change across the water-sediment interface n_{zz} will be large and thus the error will be large. To reduce this error, a very small horizontal range step must be used. However, this results in very long computer execution time. The problem of a discontinuity in sound speed at the water-sediment interface and the resultant difficulties in solving shallow water propagation problems using the SSFFT method are addressed in Jensen and Krol (1975).

Another more serious problem with the SSFFT method is that it neglects any density difference between the water and the sediment. A density difference can be important in that it influences the reflection coefficient. The problem becomes more significant as the density difference becomes larger.

In summary, the discontinuities in sound speed and in density at the water-sediment interface cause problems for the SSFFT method. The SSFFT method is therefore intrinsically better suited for deep water propagation environments for which the water-sediment interface is an unimportant feature.

C. IMPLICIT FINITE-DIFFERENCE SOLUTION METHOD

In 1979 Lee and Papadakis introduced the Crank-Nicolson implicit finite-difference method to solve the parabolic equation for underwater acoustic propagation. The Crank-Nicolson method uses a second-order central difference formula to approximate u_{zz} in (2.9) and casts the problem in the form of a tridiagonal matrix system. A representative row in the matrix system (the m^{th} row) is

$$\left(-\frac{1}{2}\frac{k}{h^2}b_m^{n+1},\ 1-\frac{1}{2}k\,a_m^{n+1}+\frac{k}{h^2}b_m^{n+1},-\frac{1}{2}\frac{k}{h^2}b_m^{n+1}\right)\begin{bmatrix}u_{m-1}^{n+1}\\u_m^{n+1}\\u_{m+1}^{n+1}\end{bmatrix} \quad 2.10$$

$$=\left(\frac{1}{2}\frac{k}{h^2}b_m^{n},\quad 1+\frac{1}{2}k\,a_m^{n}-\frac{k}{h^2}b_m^{n},\quad \frac{1}{2}\frac{k}{h^2}b_m^{n}\right)\begin{bmatrix}u_{m-1}^{n}\\u_m^{n}\\u_{m+1}^{n}\end{bmatrix}$$

where

k = horizontal range increment

h = vertical depth increment

Supercripts are used to indicate range indices and subscripts are used to indicate depth indices. In (2.10) the field is known at range index n and is to be solved at range index n + 1. Therefore, the right hand side of (2.10) reduces to a single, known value and the solution field is advanced from range index n to range index n + 1 by solving the tridiagonal system of equations.

The IFD scheme is consistant, unconditionally stable and it converges to the theoretical solution as the range and depth increments tend to zero (Lee et al., 1981). An advantage of selecting an implicit scheme over an explicit scheme is that an <u>explicit</u> scheme is only <u>conditionally</u> stable (Lee and Papadakis, 1979). Another advantage of the implicit scheme is smaller errors. More detailed

information addressing both implicit and explicit solutions of parabolic equations can be found in Gerald (1980).

The first IFD scheme handled discontinuities in the speed of sound profile but did not consider the effects of density discontinuities. It therefore did not correctly treat the interface between media having different densities.

D. IMPLICIT FINITE-DIFFERENCE METHOD: TREATMENT OF A HORIZONTAL INTERFACE

In 1982 McDaniel and Lee introduced a scheme for incorporating a horizontal interface into the IFD method. The interface separates two media with different sound speeds and densities (Figure 1).

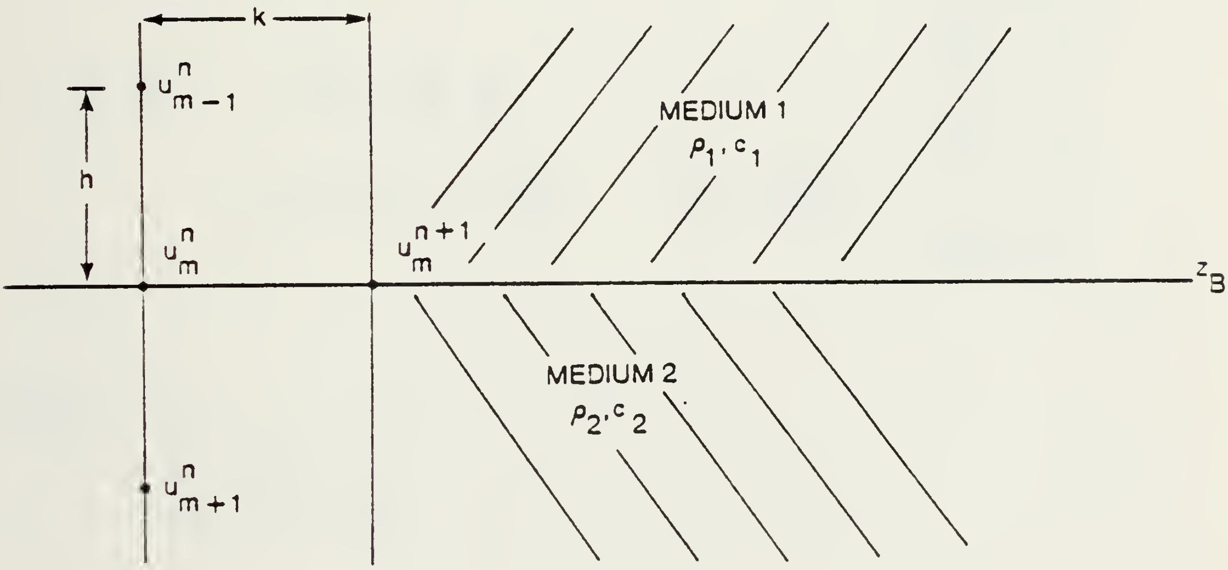


Figure 1. IFD Treatment of a Horizontal Interface

The scheme preserves continuity of pressure and continuity of the normal component of particle velocity across the interface and does not affect the stability of the IFD method.

The interface equation in the tridiagonal matrix system that results from incorporating continuity of pressure and continuity of the normal component of particle velocity is (McDaniel and Lee, 1982)

$$\begin{aligned}
&\Big(-\frac{k}{h^2} P_m^{n+1},\ 1 - \frac{1}{2} k P_m^{n+1} Q_m^{n+1} + \frac{k}{h^2} P_m^{n+1} \big(1 + \frac{\rho_1}{\rho_2}\big),\ -\frac{k}{h^2} P_m^{n+1} \frac{\rho_1}{\rho_2} \Big) \begin{bmatrix} u_{m-1}^{n+1} \\ u_m^{n+1} \\ u_{m+1}^{n+1} \end{bmatrix} \\
&= \Big(\frac{k}{h^2} P_m^{n},\ 1 + \frac{1}{2} k P_m^{n} Q_m^{n} - \frac{k}{h^2} P_m^{n} \big(1 + \frac{\rho_1}{\rho_2}\big),\ \frac{k}{h^2} P_m^{n} \frac{\rho_1}{\rho_2} \Big) \begin{bmatrix} u_{m-1}^{n} \\ u_m^{n} \\ u_{m+1}^{n} \end{bmatrix}
\end{aligned} \qquad 2.11$$

where

$$P = \left[\frac{1}{b_1} + \frac{\rho_1}{\rho_2} \frac{1}{b_2} \right]^{-1}$$

$$Q = \left[\frac{a_1}{b_1} + \frac{\rho_1}{\rho_2} \frac{a_2}{b_2} \right]$$

ρ_1 = density in layer 1 (water)

ρ_2 = density in layer 2 (sediment)

a and b are defined in (2.9)

Incorporating the horizontal interface into the IFD method requires inserting (2.10) for the row in the tridiagonal matrix system that corresponds to the interface.

The error in the solution is

$O\ (k^3 + kh)$

on the interface and

$O\ (k^3 + kh^2)$

in a continuous medium (McDaniel and Lee, 1982).

E. IMPLICIT FINITE-DIFFERENCE METHOD: TREATMENT OF A SLOPING INTERFACE

In 1983, Lee and McDaniel extended their treatment of an interface between two media to include the case of a sloping interface. As for the case of a horizontal interface, the treatment of a sloping interface preserves continuity of pressure and continuity of the normal component of particle velocity at the interface between media having different densities and sound speeds.

The problem of a sloping interface is separated into two cases: downslope, and upslope. Each case requires inserting two new rows into the IFD tridiagonal matrix system. One new row is required at the level corresponding to the interface at the range where the solution is known and the second new row is required at the level corresponding to the

interface at the range at which the solution is to be solved. Therefore, four new equations are required to cover both the downslope and upslope cases. The four sloping interface equations are derived and shown in Lee and McDaniel (1983). The equations are somewhat involved but they are of the same tridiagonal form as the original IFD matrix equations. The error for points on or adjacent to a sloping interface is (Lee and McDaniel, 1983)

$$O\ (k^3 + kh).$$

III. COMPUTER IMPLEMENTATION

A. INTRODUCTION

An IFD solution program that implements (2.10), (2.11) and the four sloping interface equations has been developed to solve underwater propagation problems. The program is written in FORTRAN using single precision, complex arithmetic and has been installed on the Naval Postgraduate School's IBM-3033 digital computer. Appendix A contains a program listing.

The solution program consists of one main program and 20 subroutines. A modular program construction was selected for flexibility and clarity.

Within each routine a structured programming approach is utilized. The structured program format, coupled with generous commenting, makes the program relatively easy to trace through.

As presently installed on the NPS computer the solution program is run interactively. Appendix B contains details.

B. GENERAL CHARACTERISTICS

The solution program handles the following environmental conditions: range independent sound speed profile in the water column, range dependent bottom profile and iso-speed, iso-density sedimentary bottom layer. The program utilizes

a Gaussian starting field and an artificial pressure release surface in the sediment at a user-specified depth. (An artificial pressure release surface is not required for the IFD solution method; however, such a surface permits efficient solution for the pressure field when it is known to tend to zero at great depths in the bottom.)

Attenuation in the water and sediment is introduced using complex indices of refraction. Artificially strong attenuation is applied in the lower portion of the sediment layer to enhance attenuation of the field above the artificial pressure release surface.

A user-specified bottom profile is input as a series of linear bottom segments. The range step along a horizontal bottom segment is set to a user-specified value. The range step along a sloping bottom segment is automatically set by the program so that the sloping bottom intersects the next vertical grid point. Bottom modifications are required in certain situations to meet the requirements that the range step not be too large and that the interface must pass through a grid point at every range at which the pressure field is solved. For a very gently sloping bottom, if the calculated range step exceeds a user-specified value then the program will automatically model the bottom as a series of level and sloping sections. For these cases, the difference between the modified bottom and the user-specified bottom is always less than or equal to one-half

the vertical grid increment. The user is informed if the bottom is modified.

It is foreseen that future enhancements will increase the program's generality. In particular, changes to allow range dependent sound speed profiles, sound speed profiles in the sediment and a user-specified starting field should be relatively simple. The modular construction of the program facilitates these types of changes.

C. MAIN PROGRAM IFD

IFD is the main program. It controls program execution and calls subroutines as appropriate.

The first executable statement in IFD calls subroutine ERRSET, a system subroutine peculiar to the NPS computer that correctly sets a variable value to zero when an underflow condition exists. Most computer systems do this automatically; however, depending on the particular system a call similar to ERRSET may be required.

IFD then calls subroutine READ to read input data, subroutine SVPW to calculate the sound speed at grid points in the water column, subroutine INITAL to initialize constants and variables, and then subroutine MATCON to calculate matrix constants. Subroutine SFIELD is then called to calculate the Gaussian starting field, followed by subroutines WRITE1 and PRINT1 which write and print output data. In the context of this program, "write" refers to

writing unformatted data into a file to be used by the plotting routine and "print" refers to writing formatted data into a file which can be sent directly to the printer.

IFD then calls subroutine NEWSEG which is the beginning of a loop that is called every time a new linear bottom segment is reached. NEWSEG calculates variables that characterize a new bottom segment. The next call is to subroutine NEWMAT which calculates matrix elements for the new bottom segment and advances the solution field one range step. IFD then enters a loop that advances the solution one range step for every pass through the loop. Inside the loop the range markers are advanced and the solution is advanced one step for the downslope, level, upslope, modified bottom downslope or modified bottom upslope situation as appropriate. In addition, the artificial attenuation mentioned earlier is applied by calling ATTENU and calls are made to WRITE2 or PRINT2 as required. Finally the range is checked to see if it has advanced to the maximum range specified for the problem. If it has, then IFD calls subroutine END which passes appropriate messages to the terminal and stops program execution.

D. SUBROUTINES

1. Subroutine READ

Subroutine READ is called by IFD to read input data from unit number NIU = 51. Input data are read in free

format and data are transferred back to main program IFD via common blocks. READ contains error checks for (1) input data insufficient and (2) the final depth in the sound speed profile unequal to the maximum depth in the water column. If either of these error conditions exists, READ issues an appropriate error message to the terminal and stops execution.

2. <u>Subroutine SVPW</u>

Subroutine SVPW calculates the vertical grid spacing used in the water and sediment. It also calculates the speed of sound at each of the grid points in the water column. Linear interpolation is used to calculate the sound speed at grid points between points on the user-specified sound speed profile.

3. <u>Subroutine INITAL</u>

Subroutine INITAL initializes constants and variables. If the user inputs 0.0 for the value of the reference sound speed then INITAL sets the reference sound speed c_o to the sound speed averaged over the deepest water column. If the user inputs 0.0 for the value of the maximum range step then INITAL sets the maximum range step to the reference wavelength,

$$\text{DRMAX} = \text{XLAMDA} \quad .$$

Setting the maximum range step to the reference wavelength is somewhat arbitrary; however, until the actual limit on the range step is better understood it serves as a rough

rule of thumb. Finally, if the user inputs 0.0 for the value of the range step along a horizontal interface then INITAL sets the range step to half of the reference wavelength,

DRLVL = 0.5 * XLAMDA.

The default range step along a horizontal interface is half the default maximum range step.

4. <u>Subroutine MATCON</u>

Subroutine MATCON calculates constants needed to compute tridiagonal matrix elements. Most of the constants computed in MATCON have no direct physical significance but contribute to computational efficiency. Attenuation in both the water and sediment is calculated with the help of a complex index of refraction n,

$$n = \left[\frac{c_o}{c_j}\right] \left(1 + i \frac{\text{BETA}}{54.575054}\right)$$

or

$$n^2 = \left[\frac{c_o}{c_j}\right]^2 + i \left[\frac{c_o}{c_j}\right]^2 \frac{\text{BETA}}{27.287527}$$

where

BETA = attenuation (dB/wavelength)

c_o = reference sound speed (m/s)

c_j = sound speed (m/s) at point j

54.575054 = conversion factor used in converting db/wavelength to nepers/meter.

5. <u>Subroutine SFIELD</u>

Subroutine SFIELD calculates the Gaussian starting field at range r = 0. This subroutine is identical with that of Lee and Botseas (1982); both yield the starting field suggested by Brock (1978),

$$U(I) = CMPLX\ (PR,\ 0.0)$$

where

$$PR = GA\ [e^{-\left[\frac{ZM-ZS}{GW}\right]^2} - e^{-\left[\frac{-ZM-ZS}{GW}\right]^2}]$$

ZM = depth (m) of grid point

ZS = source depth (m)

GW = Gaussian width (m) (= 2/FK)

FK = reference wave number (1/m) ($= 2\pi f/c_o$)

GA = Gaussian amplitude [= $(2/FK)^{1/2}/GW$

6. <u>Subroutine WRITE1</u>

Subroutine WRITE1 outputs data to a file that is used by the plotting routine. This output file corresponds to unit file number NOU = 52.

7. <u>Subroutine PRINT1</u>

Subroutine PRINT1 outputs formatted data to a file that can be sent to the printer. The output file corresponds to unit file number NPOUT = 55.

8. Subroutine NEWSEG

Subroutine NEWSEG is called at the start of each new linear bottom segment. NEWSEG computes and initializes variables that depend on characteristics of the segment. One of the variables initialized is ISLOPE which is a slope flag having value 1 if the bottom slopes down, 2 if the bottom is horizontal, 3 if the bottom slopes up, 4 if the bottom slopes down but must be modified because the slope is very small, or 5 if the bottom slopes up but must be modified because the slope is very small. NEWSEG also issues error or warning messages as appropriate.

9. Subroutine NEWMAT

Subroutine NEWMAT calculates matrix elements for the X and Y matrices. The Y matrix corresponds to the range at which the solution field is known and the X matrix corresponds to the range at which the solution field is to be found. The Y matrix is multiplied by the known solution field to obtain the right-hand side column vector needed to solve the tridiagonal system.

NEWMAT sets up the tridiagonal matrix system for the new bottom segment and than calls TRIDG to solve the system at the first range step. It then calls ATTENU to apply artificial attenuation, calls WRITE2 or PRINT2 as required, and finally updates the interface pointer that indicates the index of the grid point at the water-sediment interface.

10. <u>Subroutine WRITE2</u>

Subroutine WRITE2 is basically a continuation of subroutine WRITE1. It outputs data to a file corresponding to unit file number NOU = 52. This is the file that is used by the plotting routine. At range intervals specified by the user, WRITE2 outputs range, depth, and the value of u(r,z).

11. <u>Subroutine PRINT2</u>

Subroutine PRINT2 is basically a continuation of subroutine PRINT1. It outputs formatted data to a file corresponding to unit file number NPOUT = 55. This file can be sent directly to the printer. At range and depth intervals specified by the user PRINT2 outputs tabular values of transmission loss and u(r,z).

12. <u>Subroutine TRIDG</u>

Subroutine TRIDG solves a linear tridiagonal matrix system. TRIDG is a modified version of subroutine TRID as listed in Gerald (1980). The major modifications to subroutine TRID involved introducing arrays CTWO and CR to preserve the original matrix element values and to make the routine more efficient. Introducing the two new arrays requires more storage space but results in a substantial savings in execution time, particularly for the case of a horizontal interface.

13. <u>Subroutine TRIDL</u>

Subroutine TRIDL is a modified version of subroutine TRIDG. TRIDL differs from TRIDG in that it does not compute arrays CTWO and CR but rather uses the array values calculated in TRIDG.

14. <u>Subroutine DOWN</u>

Subroutine DOWN updates the tridiagonal matrix and calls subroutine RHS to update the right-hand side for the case of a downward sloping interface. DOWN then calls subroutine TRIDG to solve the tridiagonal system of equations and finally updates the interface pointer.

15. <u>Subroutine UP</u>

Subroutine UP performs exactly the same tasks as subroutine DOWN, but for the case of an upward sloping interface. Subroutine TRIDG is again called to solve the system.

16. <u>Subroutine LEVEL</u>

Subroutine LEVEL is called to advance the solution for the case of a horizontal interface. For this case the tridiagonal matrix elements at the advanced range need not be changed from the previous calculation. Therefore, LEVEL need only update the right-hand side by calling RHS and then solve the system by calling TRIDL.

17. <u>Subroutine RHS</u>

Subroutine RHS computes the right-hand side of the tridiagonal system by multiplying tridiagonal matrix Y by

the known solution field U(I). The resultant right-hand side column vector is stored in C(I,4).

18. <u>Subroutine</u> <u>SSLOPE</u>

Subroutine SSLOPE is called to advance the solution in the case where the bottom has been modified. SSLOPE determines which of three cases a particular section falls into: a level section following a level section, a level section following a sloping section, or a sloping section. For the case of a level section following a level section SSLOPE calls LEVEL to advance the solution. For the case of a level section following a sloping section SSLOPE updates appropriate matrix elements, calls RHS and then calls TRIDG to advance the solution. And in the case of a sloping section SSLOPE updates matrix elements and calls either DOWN or UP as appropriate.

19. <u>Subroutine</u> <u>ATTENU</u>

Subroutine ATTENU applies artificial attenuation to the bottom portion of the sediment layer as suggested by Brock (1978). The artificial attenuation matrix ATT(1) is calculated in subroutine NEWMAT.

20. <u>Subroutine</u> <u>END</u>

Subroutine END is called when the solution field has reached the maximum range specified. END sends appropriate messages to the terminal and stops execution. The messages are applicable to the program as installed on the NPS

computer but may not be appropriate for the program if installed on another system.

E. INPUT DATA

1. Input File

The input data must be stored in a file corresponding to unit number NIU as assigned in subroutine READ. In its present form READ sets the input unit number to NIU = 51. If the user prefers to read the data from a different unit (for example, a card reader), then variable NIU in READ should be set equal to the appropriate unit number.

2. Input Format

The input data is read in free format. The input card images (or input cards) are arranged as follows:

CARD	CONTENTS
1	FRQ, ZS, ZR, C0, N

where

FRQ = frequency (Hz)

ZS = source depth (m)

ZR = receiver depth (m)
(program will reset to depth of nearest grid point)

C0 = reference sound speed (m/s)
If C0 = 0.0, C0 is set to the sound speed averaged over the deepest water column.

N = number of vertical grid points

If the user desires that every integer depth value correspond to a grid point then (neglecting dimensions) N should be set to an integer multiple of ZLYR2, the depth of the pressure release surface.

<u>CARD</u> <u>CONTENTS</u>

2 RMAX, DRLVL, DRMAX, WDR, PDR, PDZ

where

RMAX = maximum range (m) of solution

DRLVL = range step (m) for marching solution along horizontal interface

If DRLVL = 0.0, then DRLVL is set to 1/2 wavelength.

If DRLVL is greater than DRMAX, then DRLVL is set to DRMAX.

DRMAX = maximum allowable range step (m)

If DRMAX = 0.0, then DRMAX is set to 1 wavelength.

WDR = range increment (m) at which solution is written to file used by plotting routine

PDR = range increment (m) at which solution is printed

PDZ = depth increment (m) rounded to nearest DZ at which solution is printed

CARD	CONTENTS	
3	BR(1) , BZ(1)	**BOTTOM PROFILE**
4	BR(2) , BZ(2)	Range and depth of water (m).
5	BR(3) , BZ(3)	Maximum number of points = 100.
.	. .	Program will reset depths to
.	. .	nearest grid point.
.	. .	
N	. .	
N+1	-1 -1	This card marks end of bottom profile.
N+2	ZLYR1, RHO1, BETA1	

where

ZLYR1 = maximum water depth (m)

RHO1 = density of water (g/cm^3)

BETA1 = attenuation in water (dB/meter)

If BETA1 is less than 0.0 then program calculates BETA1 with an empirical formula (Brock, 1978).

N+3	ZSVP(1), CSVP(1)	**SOUND SPEED PROFILE**
N+4	ZSVP(2), CSVP(2)	Depth (m) and sound speed (m/s).
.	. .	ZSVP(1) must equal 0.
.	. .	The last depth must equal
N+M		ZLYR1.

CARD	CONTENTS
N+M+1	ZLYR2, RHO2, BETA2, C2

where

ZLYR2 = depth (m) of pressure release surface at bottom of sediment layer

RHO2 = density (g/cm^3) of sediment

BETA2 = attenuation (dB/wavelength) in sediment

C2 = sound speed (m/s) in sediment

N+M+2 ZABLYR

where

ZABLYR = depth (m) of upper surface of artificial attenuation layer in sediment. ZABLYR should be about 3/4 of ZLYR2.

F. PROGRAM OUTPUT

1. Output Printer File

The program outputs formatted data to a file corresponding to unit number NPOUT which is set to 55. This formatted data file may be sent to the printer if desired. On another system the user may elect to assign NPOUT to the unit number corresponding to the printer and thereby send the formatted data directly to the printer.

2. Output Plotter File

The program outputs unformatted data to a file that is used by the plotting routine. The unit number for this

file is NOU which is set to 52. The output data in this file are stored as follows:

LINE	CONTENTS
1	RMAX

where

RMAX = maximum range (m) of solution

2	RA, ZR, U
3	RA, ZR, U
.	. . .
.	. . .

where

RA = range (m)

ZR = depth (m)

U = complex u at specified range and depth

3. Terminal Output

Certain WRITE statements in the program specify unit number 6. Unit number 6 on the NPS computer for an interactive program corresponds to terminal output. If the program is not run interactively then WRITE statements with unit number 6 may be deleted. Any pertinent information passed to the terminal during an interactive run is also passed to unit number 55.

G. PLOTTING PROGRAM

Appendix C contains a listing of the IFD plotting program installed on the NPS computer. The filename and

filetype of the program are PLOTIFD FORTRAN. This program was written separately from the IFD program because it was recognized that different computer installations have different plotting facilities. For users with different facilities the program will be a helpful reference.

The program reads data from unit number NOU = 52 which corresponds to a file with filename/filetype IFDOUT PLOTTER. Details concerning using the plotting routine are included in Appendix B.

IV. TEST CASES

A. HORIZONTAL INTERFACE CASES

The IFD program treats a horizontal interface using the same theoretical approach as the IFD program published by Lee and Botseas (1982). Throughout the remainder of this thesis the Lee and Botseas (1982) program will be called the FINITE-DIFFERENCE program and the program presented in this thesis will be called the IFD program. Two cases were run to confirm that the IFD program is in agreement with the FINITE-DIFFERENCE program for horizontal interfaces.

1. Isospeed Shallow Water

This case, first published by Jensen and Kuperman (1979), considers propagation in a shallow water, isospeed environment. The water depth is 100 meters and the solution field is calculated out to 25 kilometers. The sound speed is 1500 m/s in the water and 1550 m/s in the sediment. Density and attenuation in the sediment are 1.2 g/cm^3 and 1 dB/wavelength respectively. The source and receiver are both at 50 m and the source frequency is 500 Hz.

Solutions obtained using a normal mode program (SNAP), a split-step fast Fourier transform program (PAREQ) and the FINITE-DIFFERENCE program are shown in Figure 2. SNAP and PAREQ are programs that were developed at SACLANT Centre and are discussed further in Jensen and Kuperman

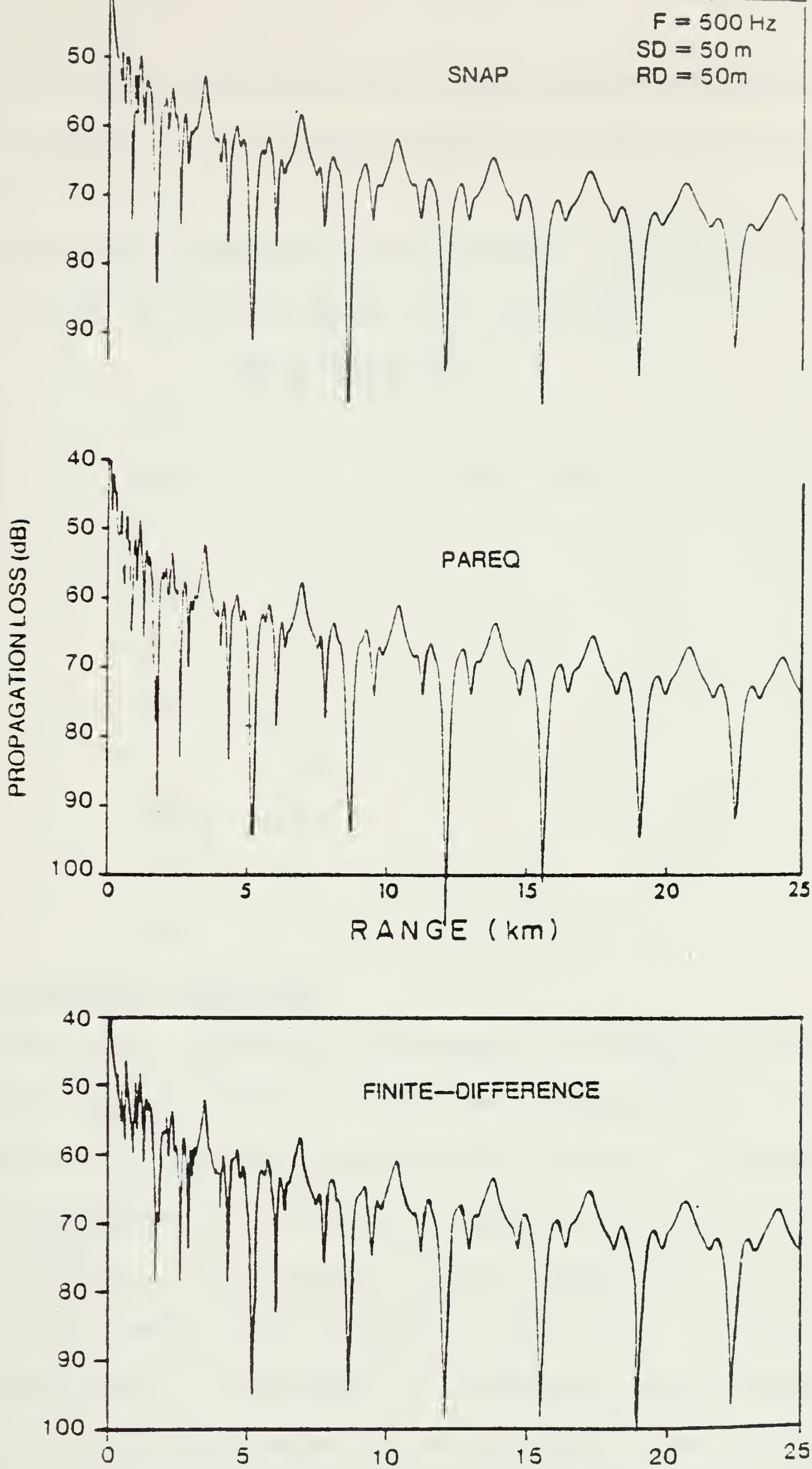


Figure 2. Propagation Loss Versus Range for Shallow Water Case; SNAP, PAREQ and FINITE-DIFFERENCE Results

(1979). The solution obtained using the IFD program is shown in Figure 3. All of the solutions are in excellent agreement.

The input runstream that produced the results shown in Figure 3 for the IFD program is as follows:

<u>Input Runstream</u>

```
500      50    50    0     500
25000    5     5     50    5000    50
0          100
25000      100
-1         -1
100      1.0   -1.0
0        1500
100      1500
250      1.2    1.0   1550
200
```

2. <u>Horizontal Interface</u>

This case, called the "horizontal interface problem" in Lee and Botseas (1982), considers propagation in an environment with the sound speed profile shown in Figure 4. Source frequency is 100 Hz, source depth is 30 m and receiver depth is 90 m. The density in the bottom is 2.1 g/cm^3 and the sound speed in the bottom is 1505 m/s. No attenuation is applied in the water or sediment using complex indices of refraction; however, artificial attenuation is applied in the lower portion of the sediment layer.

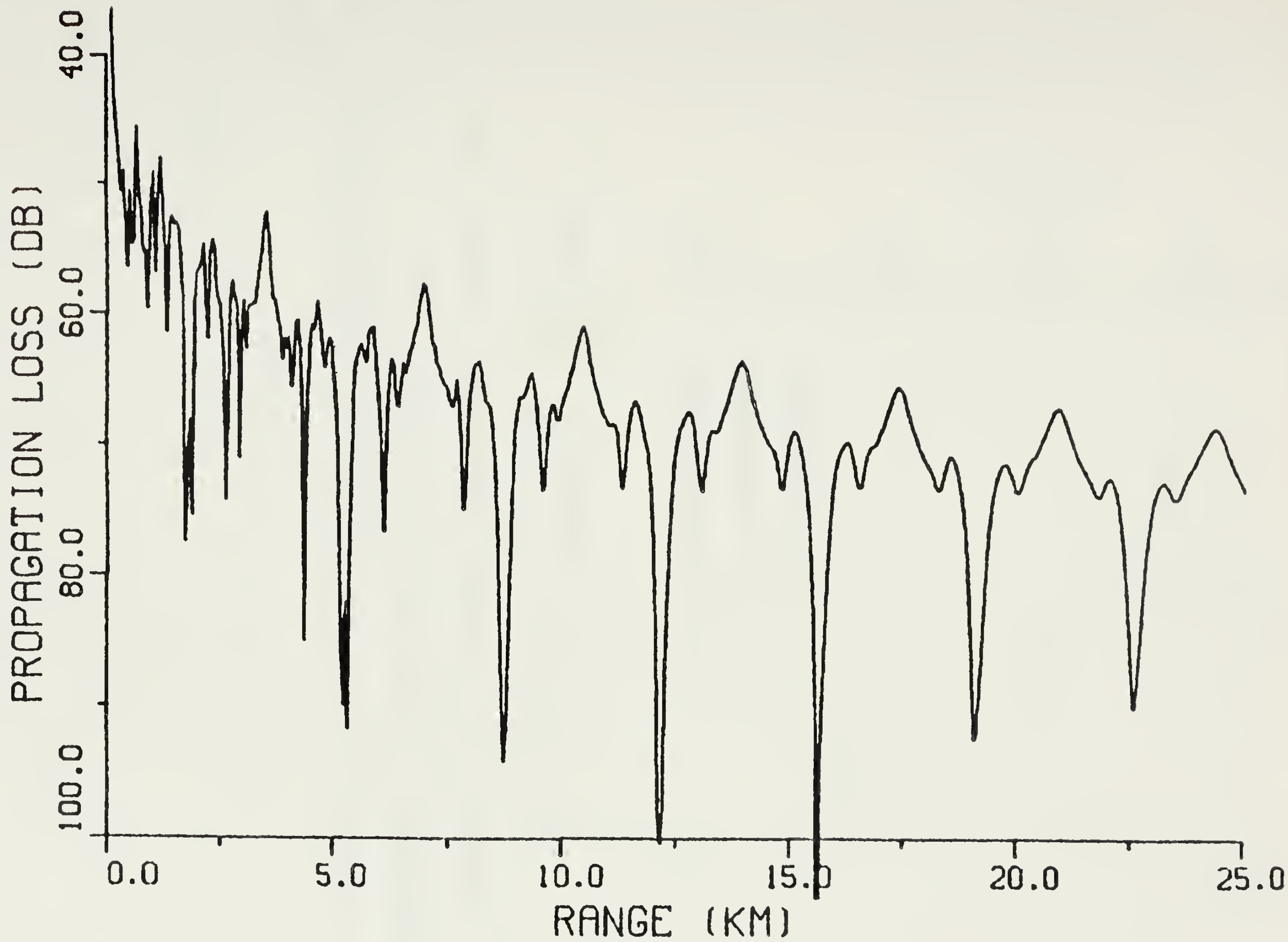


Figure 3. Propagation Loss Versus Range for Shallow Water Case; IFD Results

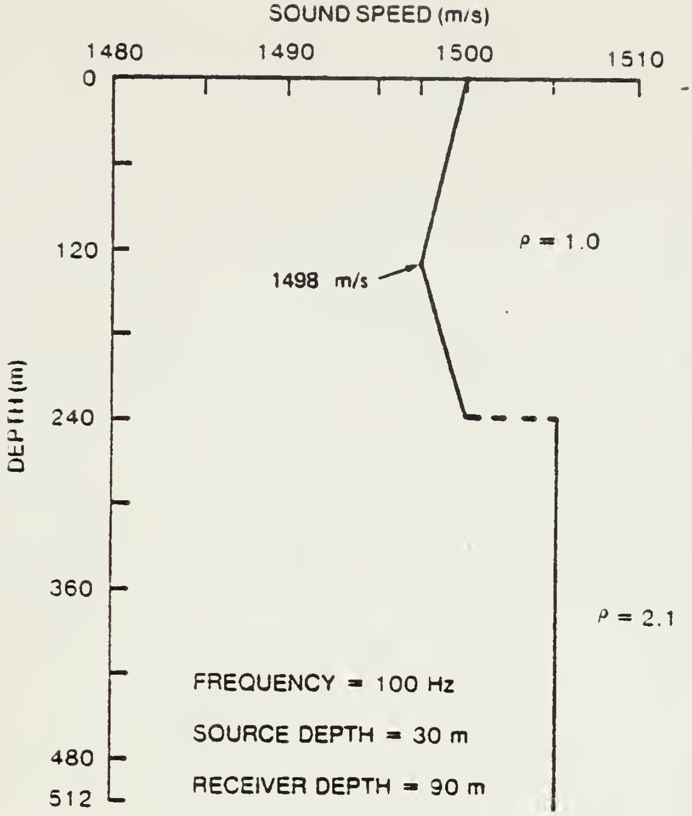


Figure 4. Horizontal Interface Case

The solution obtained using the IFD program is shown in Figure 5. This solution is in excellent agreement with the FINITE-DIFFERENCE solution shown in Lee and Botseas (1982).

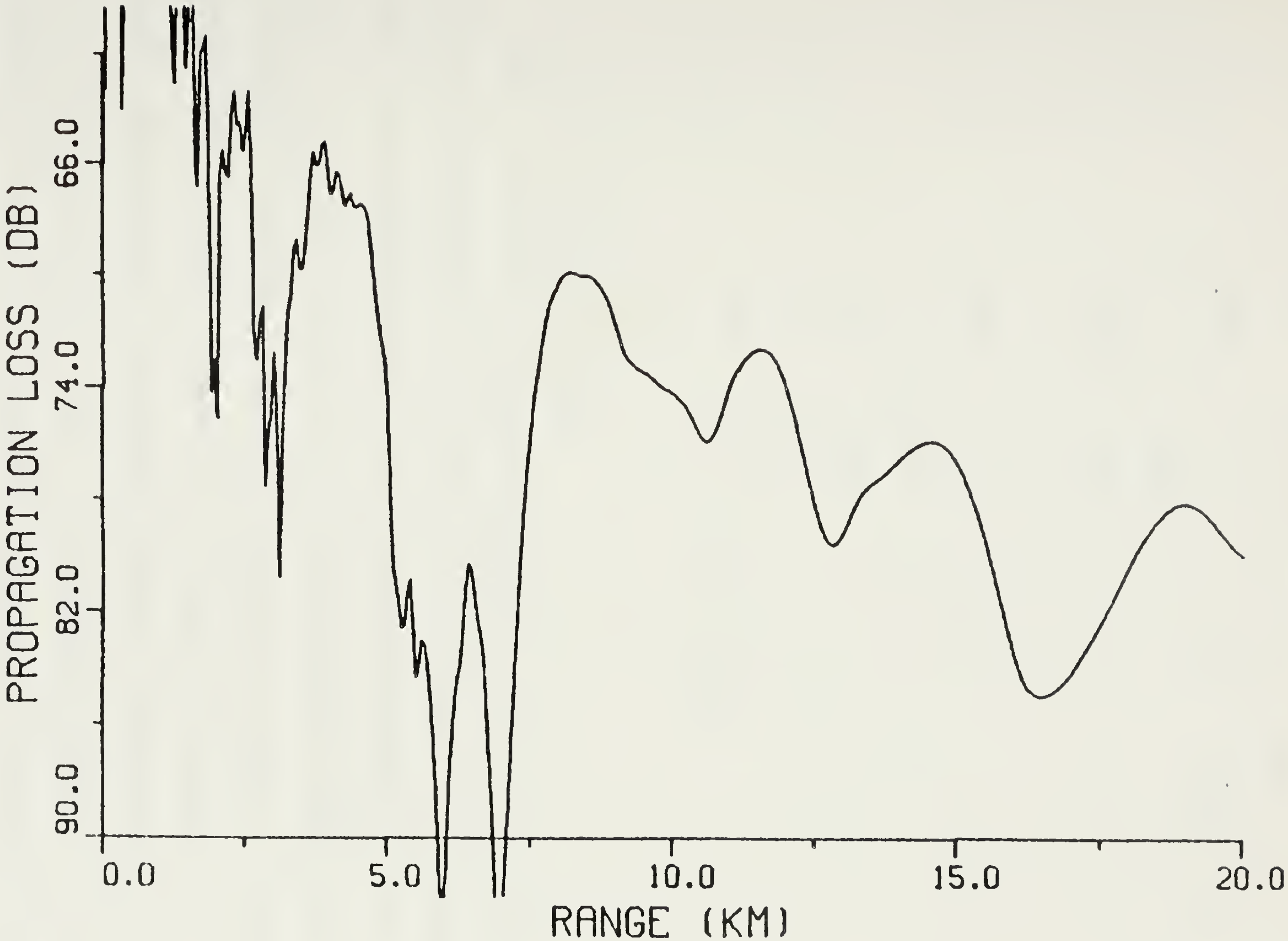


Figure 5. Propagation Loss Versus Range for Horizontal Interface Case; IFD Results

The input runstream for this case is as follows:

Input Runstream

100	30	90	0	600	
20000	2	2	50	10000	50
0	240				
20000	240				
-1	-1				
240	1.0	0.0			
0	1500				
120	1498				
240	1500				
1200	2.1	0.0	1505		
512					

B. RANGE-DEPENDENT CASES

The following range-dependent cases were solved by Jensen and Kuperman using SNAP, the normal mode program, and PAREQ, the SSFFT program (Jensen and Kuperman, 1979). The cases were also solved by Lee and Botseas using the FINITE-DIFFERENCE program (Lee and Botseas, 1982). The FINITE-DIFFERENCE program treats the sloping interface as a "stair step" and uses the interface conditions appropriate for a horizontal interface. The IFD program handles the sloping interface using the interface treatment derived by Lee and McDaniel (1983).

1. <u>Deep-to-Shallow Water</u>

This case considers propagation in an environment as depicted in Figure 6. The problem is solved for a bottom with a 8.5 degree upslope and one with a 0.85 degree upslope. Source frequency is 25 Hz, source depth and receiver depth are 25 meters. Sound speed in the water is 1500 m/s. In the sediment, sound speed is 1600 m/s, density is 1.5 g/cm^3 and attenuation is 0.2 dB/wavelength.

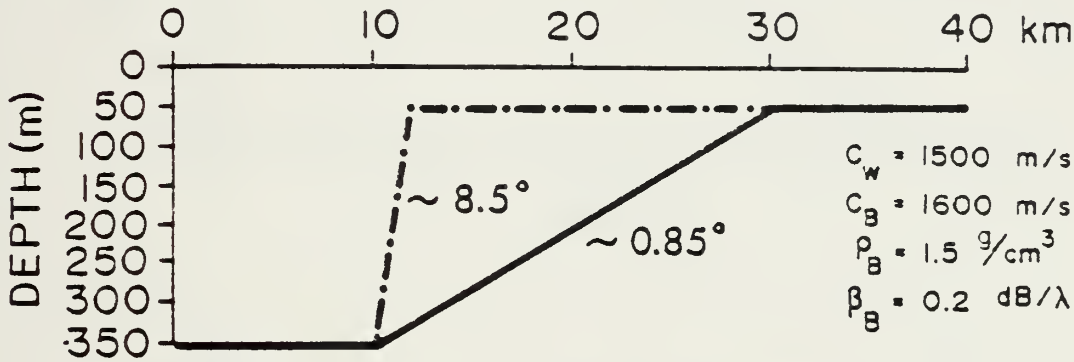
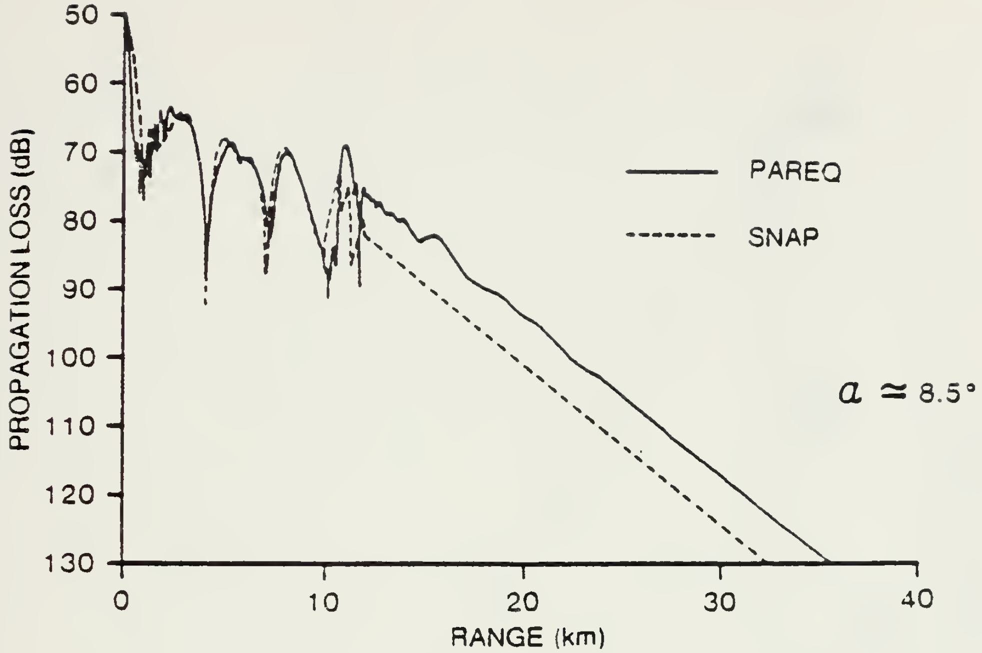


Figure 6. Deep-to-Shallow Water Case

The results for the 8.5 degree upslope case as produced by SNAP,PAREQ, and FINITE-DIFFERENCE are shown in Figure 7. The results as produced by IFD are shown in Figure 8. The difference between the results produced by SNAP and PAREQ is attributed to failure of the "adiabatic" theory underlying the SNAP program (Jensen and Kuperman, 1979). As determined from the input runstream the FINITE-DIFFERENCE results were obtained using 1.0 rather than 1.5 g/cm^3 for the density in the sediment (Lee and Botseas, 1982).



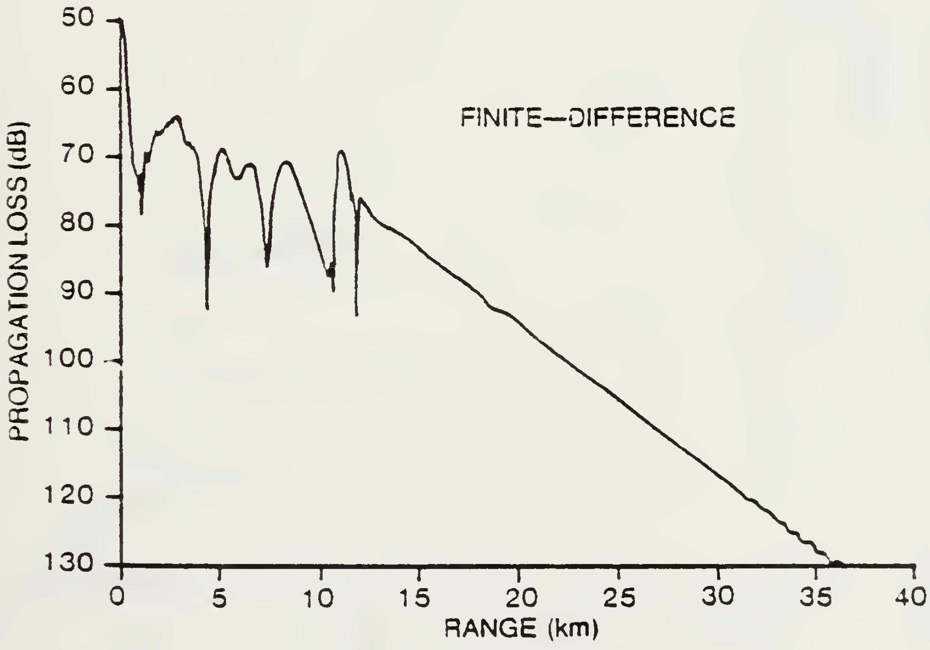


Figure 7. Propagation Loss Versus Range for Deep-to-Shallow Water Case, 8.5 Degree Slope; SNAP, PAREQ and FINITE-DIFFERENCE Results

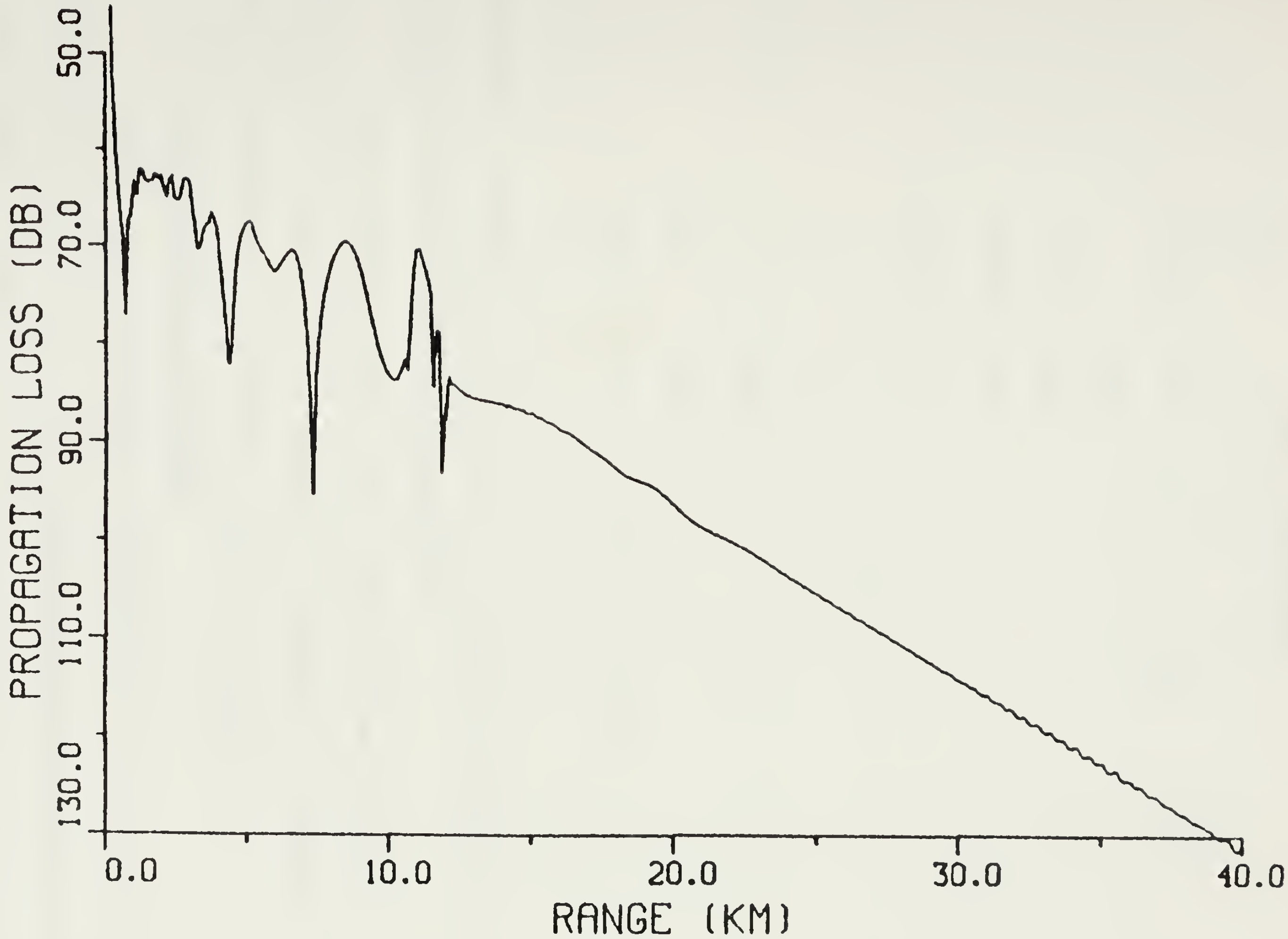


Figure 8. Propagation Loss Versus Range for Deep-to-Shallow Water Case, 8.5 degree Slope; IFD Results

The input runstream for the IFD program that produced the results shown in Figure 8 is as follows:

Input Runstream

25	25	25	0	1000	
40000	10	0	100	10000	25
0	350				
10000	350				
12000	50				
40000	50				
-1	-1				
350	1.0	-1			
0	1500				
350	1500				
1000	1.5	0.2	1600		
750					

Appendix D contains the printed output produced by the IFD program using the above runstream.

The results for the 0.85 degree upslope case as produced by SNAP and PAREQ are shown in Figure 9. The results produced by IFD are shown in Figure 10.

2. Shallow-to-Deep Water

This case considers propagation in the environment depicted in Figure 11. The environment is exactly the same as the deep-to-shallow water environment except that the shallow and deep portions have been reversed and thus the bottom slopes down rather than up.

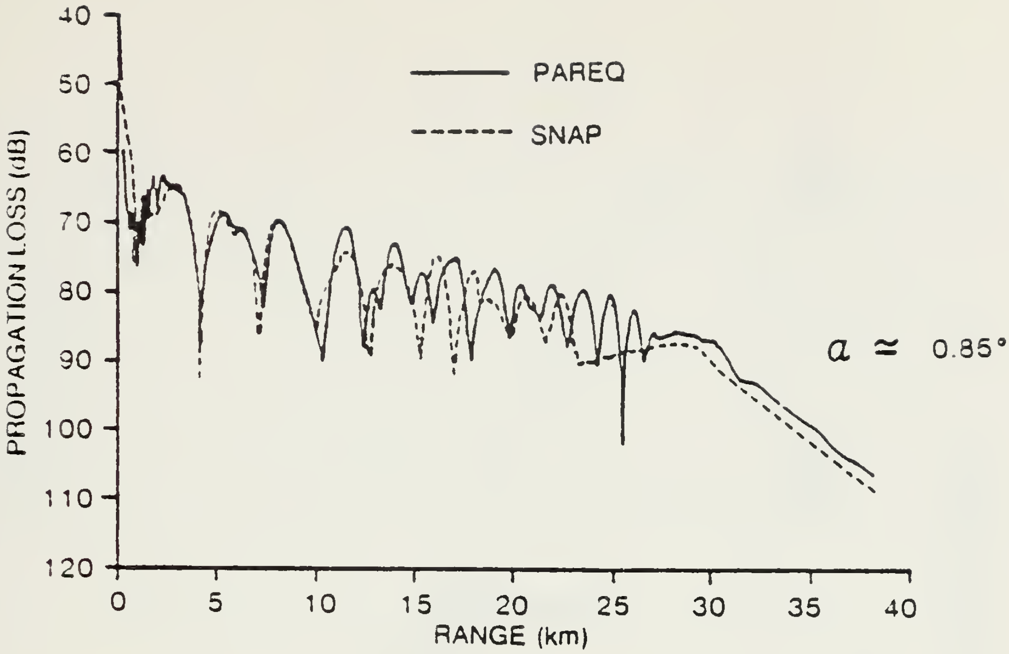


Figure 9. Propagation Loss Versus Range for Deep-to-Shallow Water Case, 0.85 Degree Slope; SNAP and PAREQ Results

The results for the 8.5 degrees downslope case as produced by SNAP and PAREQ are shown in Figure 12. As before, the difference between SNAP and PAREQ is attributed to failure of the SNAP program. The results produced by IFD are shown in Figure 13.

The results for the 0.85 degree downslope case are shown in Figures 14 and 15.

3. Comments

Differences between the results obtained using the SNAP and PAREQ programs for the range-dependent cases are discussed in Jensen and Kuperman (1979). The major differences are attributed to the violation of the adiabatic assumption in the SNAP program.

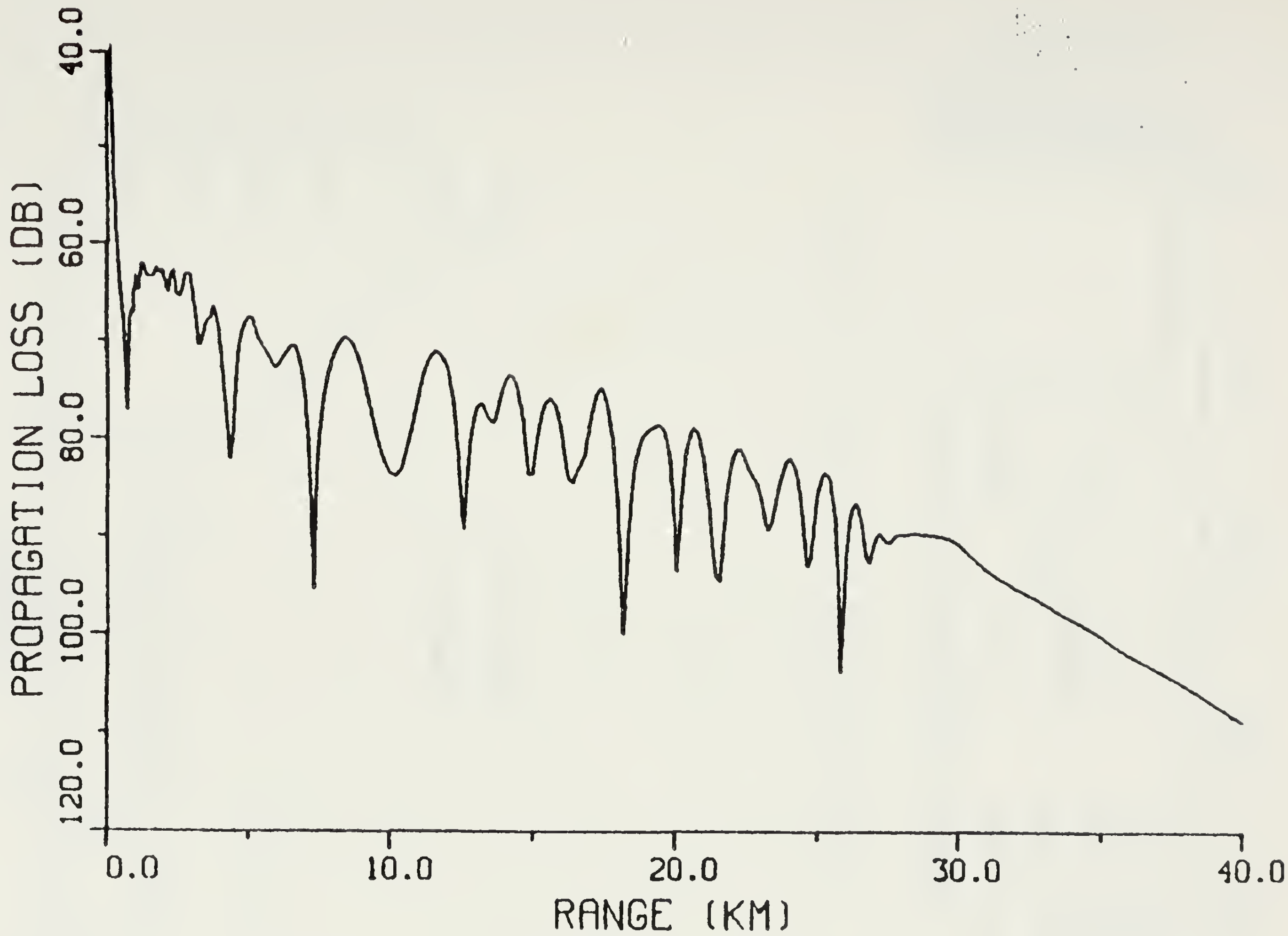


Figure 10. Propagation Loss Versus Range for Deep-to-Shallow Water Case; 0.85 degree Slope; IFD Results

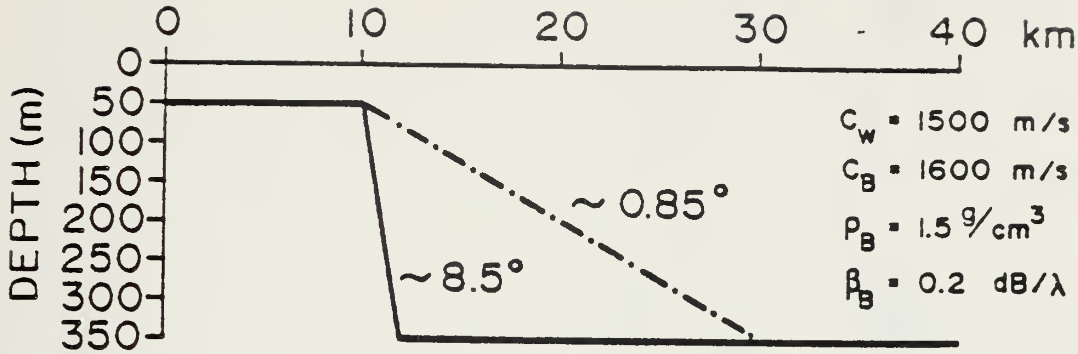


Figure 11. Shallow-to-Deep Water Case

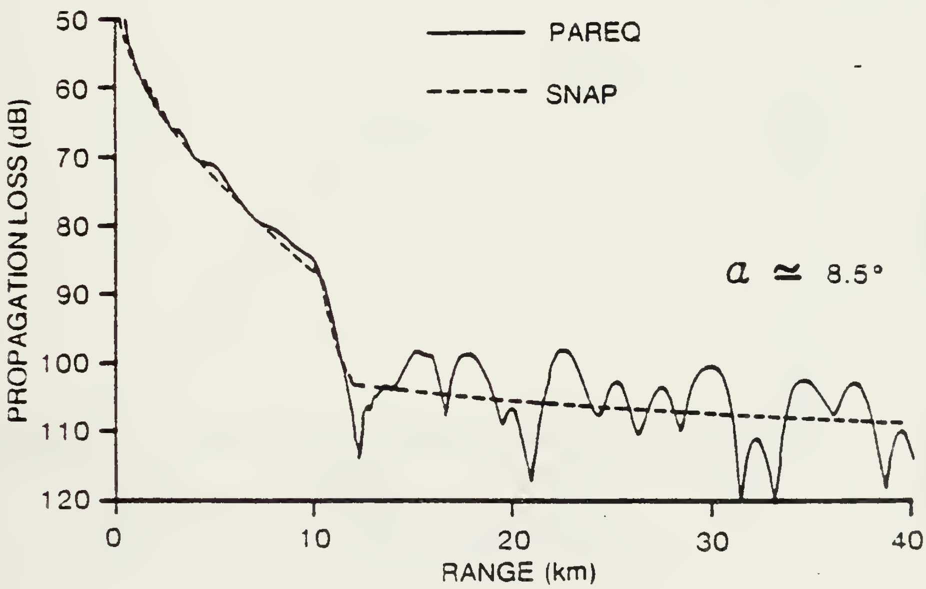


Figure 12. Propagation Loss Versus Range for Shallow-to-Deep Water Case. 8.5 degree Slope; SNAP and PAREQ Results

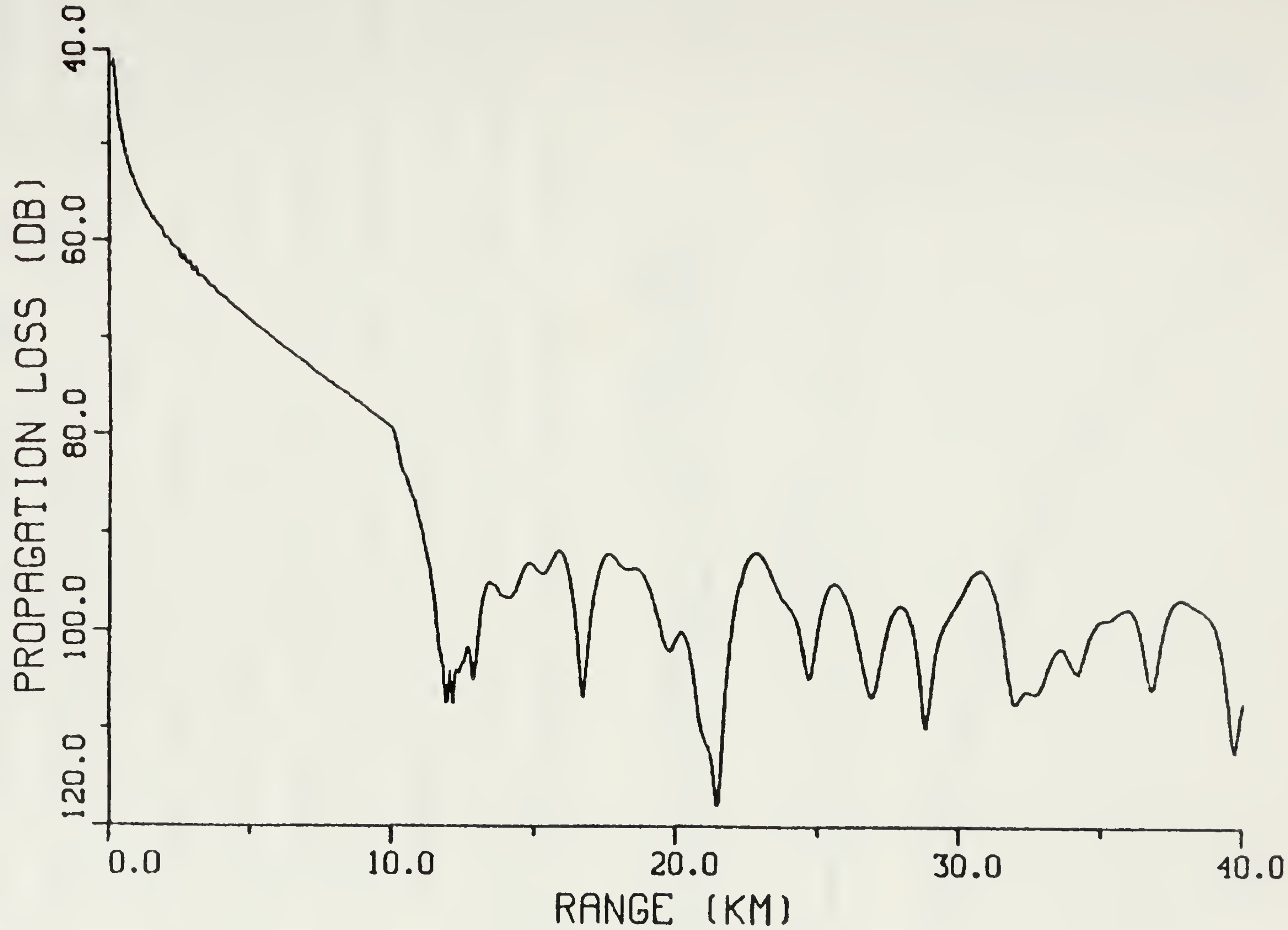


Figure 13. Propagation Loss Versus Range for Shallow-to-Deep Water Case, 8.5 degree Slope; IFD Results

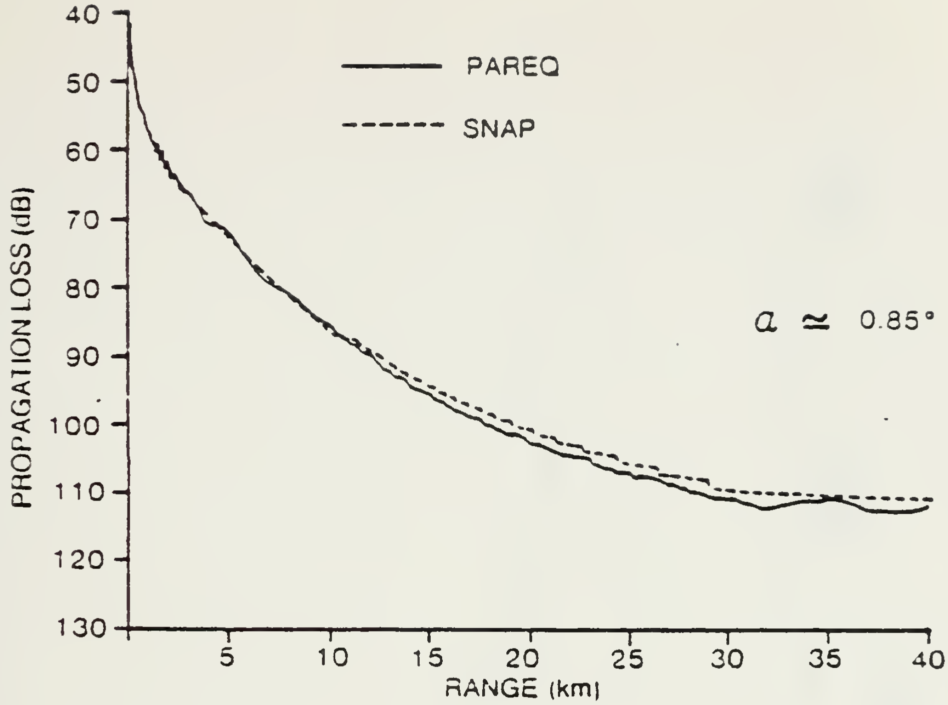


Figure 14. Propagation Loss Versus Range for Shallow-to-Deep Water Case, 0.85 degree Slope; SNAP and PAREQ Results

Figure 16 shows the results produced by IFD for the 8.5 degree, deep-to-shallow water case if 1.0 rather than 1.5 g/cm^3 is used for the density of the sediment. The IFD results obtained using 1.0 g/cm^3 are in very close agreement with the PAREQ results shown in Figure 7. However, when the correct value of 1.5 g/cm^3 is used the slope of the propagation loss curve beyond 15 km is less steep (Figure 8) and the results do not agree as well with the results produced by PAREQ. The observation that the slope of the propagation loss curve becomes less steep when 1.5 g/cm^3 is used is qualitatively consistant because a higher density difference means a higher reflection coefficient which in

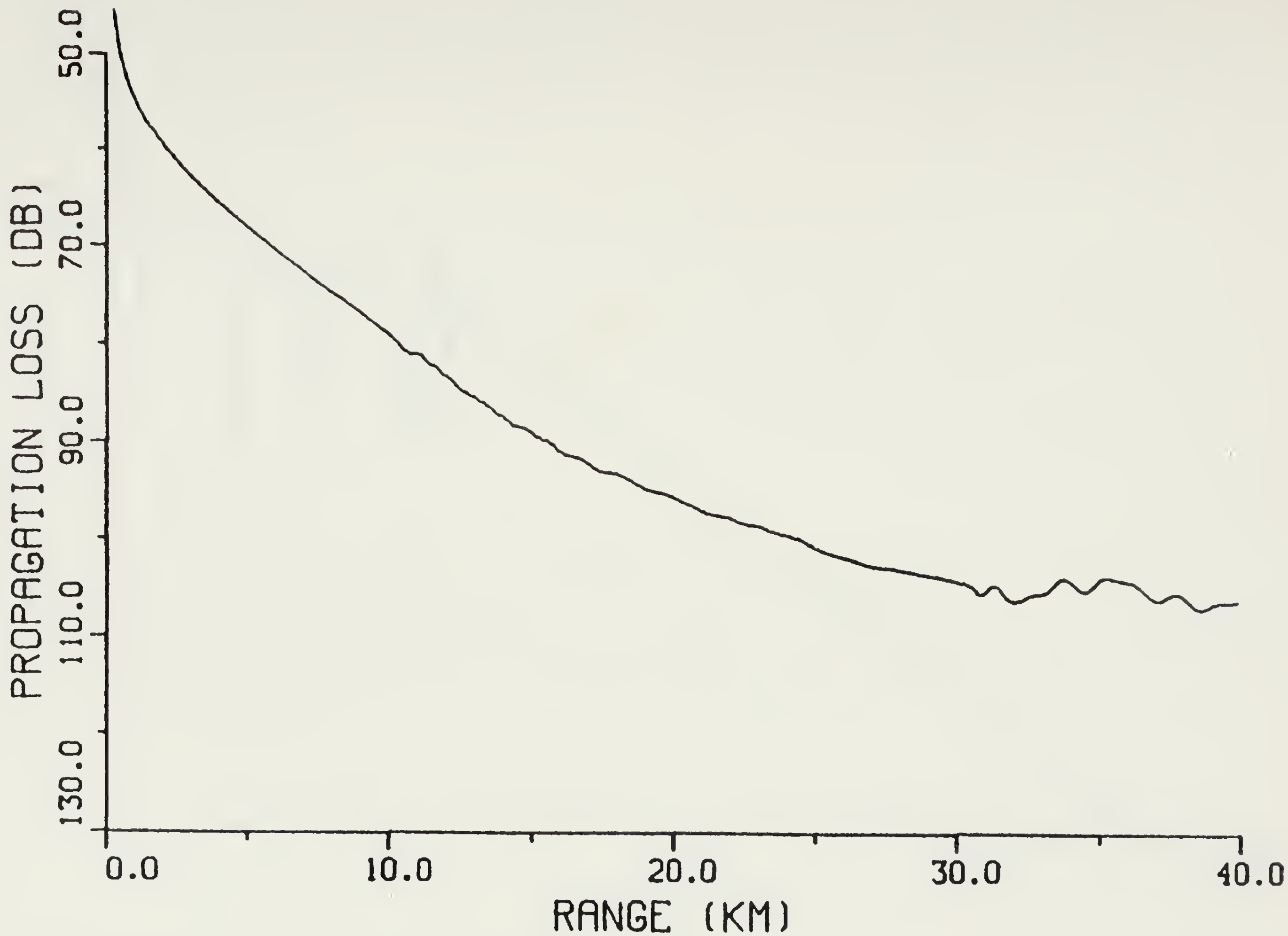


Figure 15. Propagation Loss Versus Range for Shallow-to-Deep Water Case, 0.85 degree Slope; IFD Results

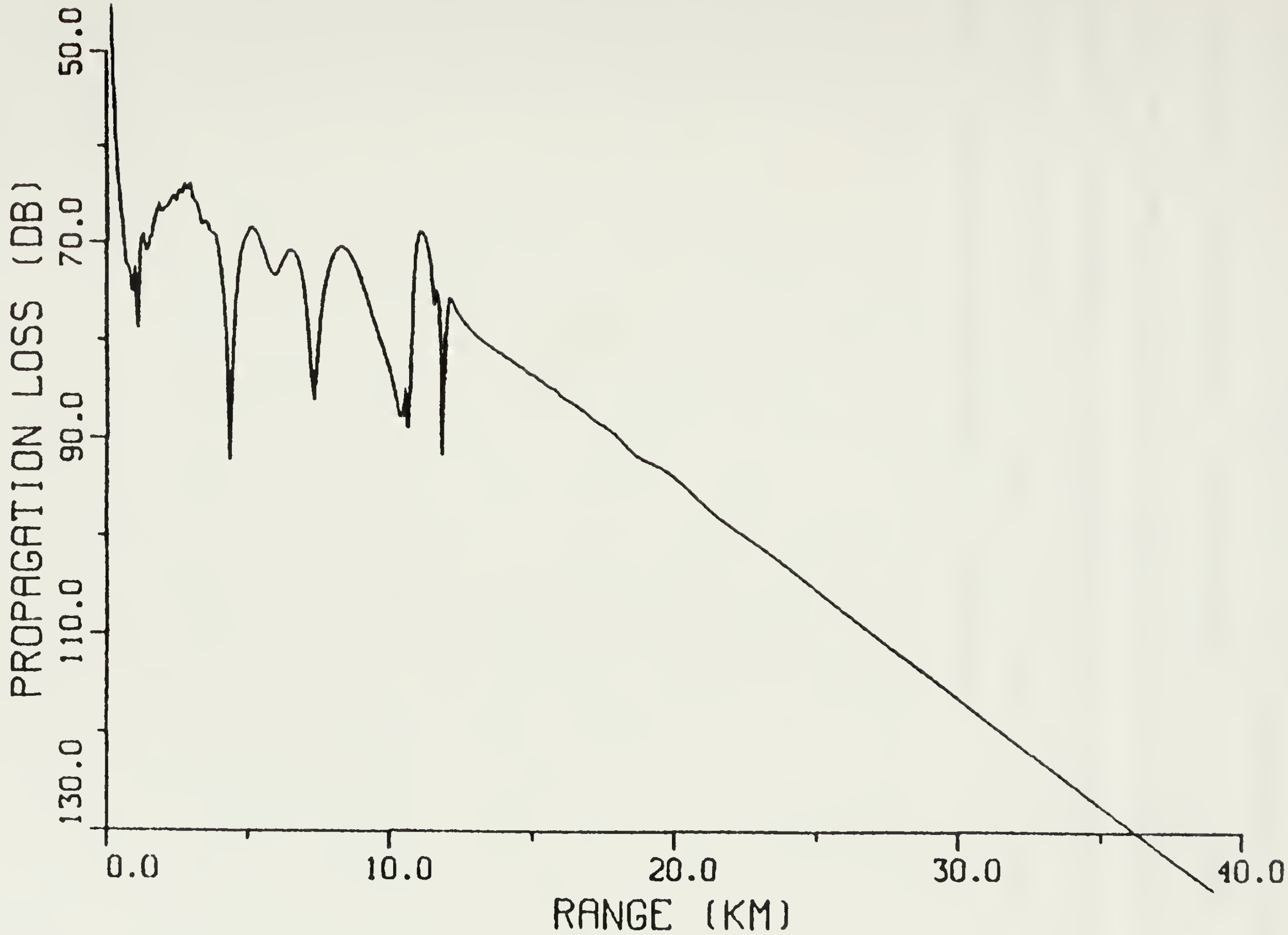


Figure 16. Propagation Loss Versus Range for Deep-to-Shallow Water Case, 8.5 degree Slope, 1.0 g/cm^3 Density in Sediment; IFD Results

turn results in more energy confined within the water layer. The difference in slope between the PAREQ and IFD results is attributed to the PAREQ program's apparent failure to account for the affects of the density discontinuity at the water-sediment interface. Because the IFD program correctly accounts for density discontinuities the results produced by IFD are believed to be more accurate than those of PAREQ when interface interaction is important.

V. COMMENTS AND CONCLUSIONS

The IFD method is an efficient, stable method for solving the parabolic equation. Use of the IFD method is particularly advantageous in shallow water environments where the water-sediment interface is an important parameter.

The IFD program presented in this thesis incorporates continuity of pressure and continuity of the normal component of particle velocity across horizontal and sloping interfaces. The program's capability to incorporate the exact interface conditions on a sloping interface, to automatically determine step-size, and to modify the bottom as required for the case of a very gently sloping bottom are important features.

Projected program enhancements include wide angle propagation (Lee and Gilbert, 1982), range-dependent sound speed profiles in the water, range-dependent sound speed profiles in the sediment layer, and multiple sediment layers with horizontal or sloping interfaces. These enhancements are listed in their approximate order of importance. The program's modular construction and structured style will facilitate implementation of these enhancements.

```
C     ******************************************************************
C     ******************************************************************
C     *   IMPLICIT FINITE-DIFFERENCE PROGRAM FOR                       *
C     *   SOLVING THE PARABOLIC EQUATION                               *
C     ******************************************************************
C     ******************************************************************
C     *   LT LARRY JAEGER                                              *
C     *   U.S. NAVAL POSTGRADUATE SCHOOL                               *
C     *   MONTEREY, CA  93943                                          *
C     ******************************************************************
C     ******************************************************************
C     ******************************************************************
C     *** ALPHABETICAL LIST OF PROGRAM VARIABLES FOLLOWS:              *
C     ******************************************************************
C     *** A      - ARRAY - COEFFICIENT A IN PARABOLIC EQUATION
C                  (IN WATER)
C     *** A2     - COEFFICIENT A IN PARABOLIC EQUATION (IN SEDIMENT)
C     *** ALPHA  - VOLUME ATTENUATION - DB/METER
C     *** ATT    - ARRAY - ATTENUATION COEFFICIENT FOR ARTIFICIAL
C                  ATTENUATION LAYER
C     *** BEDA1  - BETA 1 AS DEFINED IN LEE AND MCDANIEL (1983)
C     *** BEDA2  - BETA 2 AS DEFINED IN LEE AND MCDANIEL (1983)
C     *** BETA1  - ATTENUATION IN WATER - DB/WAVELENGTH
C     *** BETA2  - ATTENUATION IN SEDIMENT - DB/WAVELENGTH
C     *** BR     - ARRAY - RANGE FOR BOTTOM PROFILE - METERS
C     *** BZ     - ARRAY - DEPTH FOR BOTTOM PROFILE - METERS
C     *** C      - ARRAY - CONTAINS TRIDIAGONAL MATRIX SYSTEM THAT NEEDS
C                  TO BE SOLVED (SEE SUBROUTINE TRIDG)
C     *** C0     - REFERENCE SOUND SPEED - METERS/SEC
C     *** C2     - SOUND SPEED IN SEDIMENT
C     *** COSE   - COS ( THETA )
C     *** CR     - ARRAY - STORAGE SPACE USED IN SUBROUTINES TRIDG AND
C                  TRIDL
C     *** CSVP   - ARRAY - SOUND SPEED IN SOUND SPEED PROFILE -
C                  METERS/SEC
C     *** CTWO   - ARRAY - STORAGE SPACE USED IN SUBROUTINES TRIDG AND
C                  TRIDL
C     *** CWATER- ARRAY - SOUND SPEED AT GRID POINTS IN WATER COLUMN
C     *** DELTA  - DELTA AS DEFINED IN LEE AND MCDANIEL (1983)
C     *** DELIN  - 1 / DELTA
C     *** DR     - RANGE STEP - METERS
C     *** DRLVL  - RANGE STEP ALONG LEVEL INTERFACE - METERS
C     *** DRMAX  - MAXIMUM ALLOWABLE RANGE STEP - METERS
C     *** DZ     - DEPTH INCREMENT OF SOLUTION - METERS
C     *** EYE    - COMPLEX "I"
C     *** FRQ    - FREQUENCY - HZ
C     *** GAMMA1- GAMMA 1 AS DEFINED IN LEE AND MCDANIEL (1983)
C     *** GAMMA2- GAMMA 2 AS DEFINED IN LEE AND MCDANIEL (1983)
```

```
C       *** IBOT1 - POINTER THAT POINTS TO BOTTOM PROFILE POINT AT START
C                   OF BOTTOM SEGMENT
C       *** IBOT2 - POINTER THAT POINTS TO BOTTOM PROFILE POINT AT END
C                   OF BOTTOM SEGMENT
C       *** ID    - ARRAY - RUN IDENTIFICATION
C       *** IFACE - POINTER THAT POINTS TO INTERFACE AT RANGE RA1
C       *** IFACE2- POINTER THAT POINTS TO INTERFACE AT RANGE RA2
C       *** IFACEM- IFACE - 1
C       *** IFACEP- IFACE + 1
C       *** IPZ   - EVERY IPZ'TH VALUE IN DEPTH IS PRINTED
C       *** ISLOPE- SLOPE FLAG:
C                   ISLOPE = 1  -  BOTTOM SLOPES DOWN
C                   ISLOPE = 2  -  BOTTOM LEVEL
C                   ISLOPE = 3  -  BOTTOM SLOPES UP
C                   ISLOPE = 4  -  BOTTOM SLOPES DOWN, BOTTOM MODIFIED
C                   ISLOPE = 5  -  BOTTOM SLOPES UP, BOTTOM MODIFIED
C       *** ITEMP - TEMPORARY VARIABLE
C       *** IWZ   - GRID POINT CORRESPONDING TO RECEIVER DEPTH
C       *** N     - NUMBER OF EQUI-SPACED GRID POINTS IN J
C                 - INCLUDES BOTTOM POINT - DOES NOT INCLUDE SURFACE POINT
C       *** NA    - NUMBER OF POINTS IN ARTIFICIAL ATTENUATION LAYER
C       *** NBOT  - NUMBER OF POINTS IN BOTTOM PROFILE (BR AND BZ)
C       *** NIU   - UNIT NUMBER FOR INPUT DATA
C       *** NM1   - N - 1
C       *** NOU   - UNIT NUMBER FOR OUTPUT PLOTTER FILE
C       *** NPOUT - UNIT NUMBER FOR OUTPUT PRINTER FILE
C       *** NSTEP - NUMBER OF RANGE STEPS ALONG A BOTTOM SEGMENT
C       *** NSTEP1- NUMBER OF RANGE STEPS CORRESPONDING TO ONE VERTICAL
C                   GRID STEP FOR MODIFIED BOTTOM
C       *** NSVP  - NUMBER OF POINTS IN CSVP AND ZSVP
C       *** NWMAX - NUMBER OF GRID POINTS IN WATER AT MAX DEPTH
C       *** NXLFS - NUMBER OF NEXT LEVEL SECTION FOLLOWING A SLOPING
C                   SECTION FOR A MODIFIED BOTTOM
C       *** PDR   - RANGE INCREMENT AT WHICH SOLUTION IS PASSED TO
C                   OUTPUT PRINTER FILE - METERS
C       *** PDZ   - DEPTH INCREMENT AT WHICH SOLUTION IS PRINTED - METERS
C       *** PI    - THE VALUE OF PI
C       *** PL    - PROPAGATION LOSS - DB
C       *** R1    - RANGE AT WHICH BOTTOM DEPTH IS AVAILABLE - METERS
C       *** R2    - NEXT RANGE AT WHICH BOTTOM DEPTH IS AVAILABLE - METERS
C                 - INCREMENTED AS SOLUTION IS MARCHED OUT IN RANGE.
C       *** RA1   - RANGE AT WHICH SOLUTION IS KNOWN - METERS
C       *** RA2   - RANGE AT WHICH SOLUTION IS TO BE SOLVED - METERS
C                   ( RA2 = RA1 + DR )
C       *** RHO1  - DENSITY IN WATER - GM/CM**3
C       *** RHO2  - DENSITY IN SEDIMENT - GM/CM**3
C       *** RMAX  - MAXIMUM RANGE OF SOLUTION - METERS
C       *** SINE  - SIN ( THETA )
```

```
C      *** TEMP   - TEMPORARY VARIABLE                                       IFD00970
C      *** THETA  - SLOPE OF BOTTOM - RADIANS                                IFD00980
C                   THETA = 0  -  LEVEL INTERFACE                            IFD00990
C                   THETA > 0  -  INTERFACE SLOPES DOWN                      IFD01000
C                   THETA < 0  -  INTERFACE SLOPES UP                        IFD01010
C      *** U      - ARRAY - COMPLEX ACOUSTIC PRESSURE FIELD                  IFD01020
C      *** WDR    - RANGE STEP AT WHICH SOLUTION IS WRITTEN TO OUTPUT        IFD01030
C                   PLOTTER FILE                                             IFD01040
C      *** XKO    - REFERENCE WAVE NUMBER                                    IFD01050
C      *** XLI    - MATRIX ELEMENT, X MATRIX, LOWER DIAGONAL, ON INTERFACEIFD01060
C      *** XLIZ   - XLI FOR SLOPING BOTTOM                                   IFD01070
C      *** XLAMDA- REFERENCE WAVELENGTH - METERS                             IFD01080
C      *** XLRWS  - MATRIX ELEMENT, X MATRIX, OFF-DIAGONAL, IN WATER AND     IFD01090
C                   SEDIMENT                                                 IFD01100
C      *** XMI    - MATRIX ELEMENT, X MATRIX, MAIN DIAGONAL, ON INTERFACE    IFD01110
C      *** XMS    - MATRIX ELEMENT, X MATRIX, MAIN DIAGONAL, IN SEDIMENT     IFD01120
C      *** XMW    - ARRAY - MATRIX ELEMENT, X MATRIX, MAIN DIAGONAL, IN      IFD01130
C                   WATER                                                    IFD01140
C      *** XN     - REAL INDEX OF REFRACTION                                 IFD01150
C      *** XN1    - COMPLEX INDEX OF REFRACTION SQUARED                      IFD01160
C      *** XPR    - RANGE AT WHICH SOLUTION IS PRINTED - METERS              IFD01170
C      *** XWR    - RANGE AT WHICH SOLUTION IS WRITTEN TO OUTPUT             IFD01180
C                   PLOTTER FILE                                             IFD01190
C      *** XX...  - VARIABLES THAT BEGIN WITH XX HAVE NO SPECIAL PHYSICAL    IFD01200
C                   SIGNIFICANCE BUT THEY CONTRIBUTE TO COMPUTATIONAL        IFD01210
C                   EFFICIENCY.  ALL XX VARIABLES ARE CALCULATED IN          IFD01220
C                   SUBROUTINE INITAL, ALL ARE INDEPENDENT OF RANGE STEP     IFD01230
C                   AND INTERFACE SLOPE, AND ALL ARE USED TO CALCULATE       IFD01240
C                   MATRIX ELEMENTS.                                         IFD01250
C      *** YLI    - MATRIX ELEMENT, Y MATRIX, LOWER DIAGONAL, ON INTERFACEIFD01260
C      *** YLIV   - YLI FOR LEVEL INTERFACE                                  IFD01270
C      *** YLIZ   - YLI FOR SLOPING INTERFACE                                IFD01280
C      *** YLRWS  - MATRIX ELEMENT, Y MATRIX, OFF-DIAGONAL, IN WATER AND     IFD01290
C                   SEDIMENT                                                 IFD01300
C      *** YMI    - MATRIX ELEMENT, Y MATRIX, MAIN DIAGONAL, ON INTERFACE    IFD01310
C      *** YMS    - MATRIX ELEMENT, Y MATRIX, MAIN DIAGONAL, IN SEDIMENT     IFD01320
C      *** YMW    - ARRAY - MATRIX ELEMENTS, Y MATRIX, MAIN DIAGONAL, IN     IFD01330
C                   WATER                                                    IFD01340
C      *** YRI    - MATRIX ELEMENT, Y MATRIX, UPPER DIAGONAL, ON INTERFACEIFD01350
C      *** YRIV   - YRI FOR LEVEL INTERFACE                                  IFD01360
C      *** YRIZ   - YRI FOR SLOPING INTERFACE                                IFD01370
C      *** Z1     - DEPTH OF WATER AT RANGE R1 - METERS                      IFD01380
C      *** Z2     - DEPTH OF WATER AT RANGE R2 - METERS                      IFD01390
C      *** ZABLYR- DEPTH OF UPPER EDGE OF ARTIFICIAL ATTENUATION             IFD01400
C                   LAYER - METERS                                           IFD01410
C      *** ZI     - DEPTH OF GRID POINT - METERS                             IFD01420
C      *** ZLYR1  - MAXIMUM WATER DEPTH - METERS                             IFD01430
C      *** ZLYR2  - DEPTH OF PRESSURE RELEASE SURFACE - METERS               IFD01440
```

```
C     *** ZR     - RECEIVER DEPTH - METERS                               IFD01450
C     *** ZS     - SOURCE DEPTH - METERS                                 IFD01460
C     *** ZSVP   - ARRAY - DEPTH FOR SOUND SPEED PROFILE - METERS        IFD01470
C     *** ZZ...  - VARIABLES THAT BEGIN WITH ZZ HAVE NO SPECIAL PHYSICAL IFD01480
C                  SIGNIFICANCE BUT THEY CONTRIBUTE TO COMPUTATIONAL     IFD01490
C                  EFFICIENCY.  ALL ZZ VARIABLES ARE CALCULATED IN       IFD01500
C                  SUBROUTINE NEWMAT, ALL DEPEND ON RANGE STEP AND/OR    IFD01510
C                  INTERFACE SLOPE, AND ALL ARE USED TO CALCULATE MATRIX IFD01520
C                  ELEMENTS.                                             IFD01530
C     ******************************************************************IFD01540
C     *** INPUT                                                          IFD01550
C     ******************************************************************IFD01560
C     INPUT UNIT NUMBER = NIU                                            IFD01570
C     INPUT FILENAME AND FILETYPE = IFDIN DATAIN                         IFD01580
C     CONTENTS:  CARD IMAGES IN FREE FORMAT                              IFD01590
C        CARD 1     :   FRQ, ZS, ZR, CO, N                               IFD01600
C        CARD 2     :   RMAX, DRLVL, DRMAX, WDR, PDR, PDZ                IFD01610
C        CARD 3     :   BR(1), BZ(1)      **                             IFD01620
C        CARD 4     :   BR(2), BZ(2)        *    BOTTOM PROFILE          IFD01630
C          .            .                  **   RANGE, WATER DEPTH (METERS)IFD01640
C          .            .                   *                            IFD01650
C        CARD N     :   .                 **                             IFD01660
C        CARD N+1   :   -1, -1                                           IFD01670
C        CARD N+2   :   ZLYR1, RHO1, BETA1                               IFD01680
C        CARD N+3   :   ZSVP(1),CSVP(1)     **                           IFD01690
C        CARD N+4   :   ZSVP(2),CSVP(2)       *   SOUND SPEED PROFILE    IFD01700
C          .            .       .            **   DEPTH (METERS) AND SOUND IFD01710
C          .            .       .             *   SPEED (METERS/SEC)     IFD01720
C        CARD N+M   :   ZSVP(J),CSVP(J)     **                           IFD01730
C        CARD N+M+1:    ZLYR2, RHO2, BETA2, C2                           IFD01740
C        CARD N+M+2:    ZABLYR                                           IFD01750
C                                                                        IFD01760
C           WHERE:                                                       IFD01770
C              FRQ      = FREQUENCY (HZ)                                 IFD01780
C              ZS       = SOURCE DEPTH (M)                               IFD01790
C              ZR       = RECEIVER DEPTH (M)                             IFD01800
C              CO       = REFERENCE SOUND SPEED (M/S). IF CO = 0.0, CO IS IFD01810
C                         SET TO AVERAGE SOUND SPEED IN WATER COLUMN.    IFD01820
C              N        = NUMBER OF GRID POINTS                          IFD01830
C                                                                        IFD01840
C              RMAX     = MAXIMUM RANGE (M) OF SOLUTION                  IFD01850
C              DRLVL    = RANGE STEP (M) ALONG LEVEL INTERFACE.  IF      IFD01860
C                         DRLVL = 0.0, DRLVL SEY TO 1/2 WAVELENGTH.  IF  IFD01870
C                         DRLVL GREATER THAN DRMAX, DRLVL SET TO DRMAX.  IFD01880
C              DRMAX    = MAXIMUM ALLOWABLE RANGE STEP (M).  IF DRMAX = 0.0,IFD01890
C                         DRMAX IS SET TO ONE WAVELENGTH.                IFD01900
C              WDR      = RANGE STEP (M) AT WHICH SOLUTION IS  WRITTEN TO IFD01910
C                         FILE USED BY PLOTTING ROUTINE.                 IFD01920
```

```
C                ROUNDED TO NEAREST DR.
C         PDR    = RANGE STEP (M) AT WHICH SOLUTION IS PRINTED.
C                  ROUNDED TO NEAREST DR.
C         PDZ    = DEPTH INCREMENT AT WHICH SOLUTION IS PRINTED.
C                  ROUNDED TO NEAREST DZ.
C
C         BR     = RANGE (M) OF BOTTOM PROFILE
C         BZ     = DEPTH (M) OF BOTTOM PROFILE
C
C         ZLYR1  = MAXIMUM WATER DEPTH (M)
C         RHO1   = DENSITY IN WATER (GM/CM**3)
C         BETA1  = ATTENUATION (DB/WAVELENGTH) IN WATER.  IF BETA1
C                  NEGATIVE THEN ATTENUATION COMPUTED.
C
C         ZSVP   = DEPTH (M) ARRAY FOR SOUND SPEED PROFILE
C         CSVP   = SOUND SPEED (M/S) ARRAY FOR SOUND SPEED PROFILE
C
C         ZLYR2  = DEPTH (M) OF PRESSURE RELEASE SURFACE
C         RHO2   = DENSITY (GM/CM**3) IN SEDIMENT
C         BETA2  = ATTENUATION (DB/WAVELENGTH) IN SEDIMENT
C         C2     = SOUND SPEED (M/S) IN SEDIMENT
C
C         ZABLYR = DEPTH (M) OF UPPER SURFACE OF ARTIFICIAL
C                  ATTENUATION LAYER
C
C*********************************************************************
C*** OUTPUT
C*********************************************************************
C     OUTPUT UNIT NUMBER FOR FILE USED BY PLOTTING ROUTINE = NOU
C     FILENAME AND FILETYPE FOR OUTPUT PLOTTER FILE = IFDOUT PLOTTER
C
C     OUTPUT UNIT NUMBER FOR PRINT FILE = NPOUT
C     FILENAME AND FILETYPE FOR OUTPUT PRINTER FILE = IFDOUT PRINTER
C*********************************************************************
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NMI,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
```

```
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),                 IFD02410
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,             IFD02420
     *        U(5C00),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10                                 IFD02430
C                                                                              IFD02440
C     *** THE FOLLOWING CALL IS REQUIRED FOR THE NPS IBM COMPUTER.             IFD02450
C     *** IT PERMITS CONTINUED EXECUTION FOR ANY NUMBER OF UNDERFLOW           IFD02460
C     *** "ERRORS"                                                             IFD02470
          CALL ERFSET (208,300,-1,1,1)                                         IFD02480
C                                                                              IFD02490
      CALL READ                                                                IFD02500
      CALL SVPW                                                                IFD02510
      CALL INITAL                                                              IFD02520
      CALL MATCON                                                              IFD02530
      CALL SFIELD (FRQ,CO,ZS,N,DZ,U)                                           IFD02540
      CALL WRITE1                                                              IFD02550
      CALL PRINT1                                                              IFD02560
 10   CALL NEWSEG                                                              IFD02570
      CALL NEWMAT                                                              IFD02580
      IF ( NSTEP.EQ.1 ) GO TO 10                                               IFD02590
C                                                                              IFD02600
C     *** THIS LOOP ADVANCES THE SOLUTION ALONG ONE ENTIRE LINEAR              IFD02610
C     *** BOTTOM SEGMENT                                                       IFD02620
          DO 80 ISTEP=2,NSTEP                                                  IFD02630
C             *** UPDATE RANGES                                                IFD02640
                 RA1 =  R1 + (ISTEP-1)*DR                                      IFD02650
                 RA2 =  RA1 + DR                                               IFD02660
C             *** IS SEGMENT SLOPING DOWN, LEVEL, UP, MODIFIED DOWN,           IFD02670
C             *** OR MODIFIED UP?                                              IFD02680
                 GO TO (20,30,40,50,60), ISLOPE                                IFD02690
C                *** BOTTOM SLOPES DOWN                                        IFD02700
 20                  CALL DOWN                                                 IFD02710
                     GO TO 70                                                  IFD02720
C                *** BOTTOM IS LEVEL                                           IFD02730
 30                  CALL LEVEL                                                IFD02740
                     GO TO 70                                                  IFD02750
C                *** BOTTOM SLOPES UP                                          IFD02760
 40                  CALL UP                                                   IFD02770
                     GO TO 70                                                  IFD02780
C                *** BOTTOM SLOPES DOWN SLOWLY, BOTTOM MODIFIED                IFD02790
 50                  CALL SSLOPE                                               IFD02800
                     GO TO 70                                                  IFD02810
C                *** BOTTOM SLOPES UP SLOWLY, BOTTOM MODIFIED                  IFD02820
 60                  CALL SSLOPE                                               IFD02830
C                                                                              IFD02840
 70           CONTINUE                                                         IFD02850
              CALL ATTENU(U,ATT,IA,NA)                                         IFD02860
              RA2P = RA2+0.5                                                   IFD02870
C             *** TIME TO WRITE?                                               IFD02880
```

```
            IF (RA2P.GE.XWR) CALL WRITE2
C       *** TIME TO PRINT?
            IF (RA2P.GE.XPR) CALL PRINT2
C       *** TIME TO TERMINATE?
            IF (RA2.GE.RMAX) GO TO 90
 80     CONTINUE
C
C    *** GO BACK AND CONTINUE WITH NEXT LINEAR BOTTOM SEGMENT
        GO TC 1C
C
C    *** TIME TO TERMINATE
 90     CALL ENC (RA2)
     STOP
     END
C
```

```
      SUBROUTINE READ
C
C        (1)  THIS SUBROUTINE READS ALL INPUT DATA.
C        (2)  THE DATA IS READ FROM INPUT UNIT NUMBER:  NIU = 51
C        (3)  INPUT FILENAME AND FILETYPE ARE:  IFDIN  DATAIN
C        (4)  DATA IS READ IN FREE FORMAT.
C        (5)  DATA IS TRANSFERRED BACK TO MAIN PROGRAM VIA COMMON BLOCK
C
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *            NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *            CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *            R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *            XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      DATA NIU/51/, NPOUT/55/
C
C     *** READ INPUT PARAMETERS
          READ(NIU,*,END=100) FRQ, ZS, ZR, CO, N
          READ(NIU,*,END=100) RMAX, DRLVL, DRMAX, WDR, PDR, PDZ
C
C     *** READ BOTTOM PROFILE - RANGE, DEPTH
          DO 10 I=1,101
              READ(NIU,*,END=100) BR(I), BZ(I)
              NBOT=I
C             *** END OF PROFILE?
                  IF(BR(I).LT.0.0) GO TO 20
C             *** NO
 10       CONTINUE
C
 20   CONTINUE
C     *** EXTEND LAST DEPTH BEYOND MAX RANGE
          BR(NBOT) = 1.0E+10
          BZ(NBOT) = BZ(NBOT-1)
C
C     *** FIRST LAYER IS WATER. SECOND IS SEDIMENT.
C     *** READ MAX DEPTH, DENSITY AND ATTENUATION OF FIRST LAYER
          READ(NIU,*,END=100) ZLYR1, RHO1, BETA1
C
C     *** READ SOUND SPEED PROFILE IN FIRST LAYER
          DO 25 I=1,101
              NSVP=I
              READ(NIU,*,END=100) ZSVP(I), CSVP(I)
C             *** READ ANOTHER PROFILE POINT?
                  IF(ZSVP(I).LT.ZLYR1) GO TO 25
C             *** NO
C             *** WAS THAT THE LAST PROFILE POINT?
                  IF(ZSVP(I).EQ.ZLYR1) GO TO 30
C             *** NO, THERE IS ERROR.
```

```
            GO TO 101
 25      CONTINUE
C
C     *** DOES THE SOUND SPEED PROFILE START AT THE SURFACE?
 30      IF ( ZSVP(1).NE.0.0 ) GO TO 102
C     *** YES
C
C     *** READ DEPTH, DENSITY, ATTENUATION AND SPEED IN SECOND LAYER
         READ(NIL,*,END=100) ZLYR2, RHO2, BETA2,C2
C     *** REAC DEFTH OF UPPER EDGE OF ARTIFICIAL ATTENUATING LAYER
         READ(NIU,*,END=100) ZABLYR
      RETURN
C
C     *** ERROR EXITS
 100  WRITE(6,900)
      WRITE(NPOUT,900)
      STOP
 101  WRITE(6,901)
      WRITE(NPOUT,901)
      STOP
 102  WRITE(6,902)
      WRITE(NPOUT,902)
      STOP
C
 900  FORMAT(//,1X,'ERROR:  EXPECTING MORE INPUT CATA. ',/, 9X,
     *       'EXECLTION TERMINATED.',//)
 901  FORMAT(//,' ERROR:  FINAL DEPTH IN SOUNC SPEED PROFILE DOES NOT '
     *       ,/,9X,'EQUAL MAXIMUM DEPTH OF WATER CCLUMN.',//,9X,
     *       'EXECLTION TERMINATED.',//)
 902  FORMAT(//,' ERROR:  FIRST DEPTH IN SOUNC SPEED PROFILE'
     *       ,/,9X,'DOES NOT EQUAL ZERO.',//,9X,
     *       'EXECLTION TERMINATED.',//)
C
      END
```

```
      SUBROUTINE SVPW
C
C         (1)  THIS SUBROUTINE CALCULATES THE VERTICAL STEP SIZE:  DZ
C         (2)  THIS SUBROUTINE ALSO CALCULATES THE SPEED OF SOUND AT
C              EACH OF THE VERTICAL GRID POINTS.
C         (3)  SOUND SPEED VALUES ARE DETERMINED BY LINEAR INTERPOLATION.
C         (4)  SOUND SPEEDS ARE STORED IN CWATER(I).
C              (A)  THE INDEX I RANGES FROM 1 TO NWMAX.
C              (B)  CWATER(1) CORRESPONDS TO THE GRID POINT DZ BELOW
C                   THE SURFACE.
C
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *            NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),C0,
     *            CSVF(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *            R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *            XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
C
C     ***CALCULATE VERTICAL STEP SIZE
         DZ = ZLYR2 / FLOAT(N)
C
C     ***CALCULATE NUMBER OF GRID POINTS IN WATER COLUMN
         NWMAX = INT((ZLYR1/DZ)+0.5)
C
C     ***CALCULATE SOUND SPEED AT ALL GRID POINTS IN WATER COLUMN
         L = 1
         DO 20 I=1,NWMAX
            ZI = I*DZ
            LP1 = L+1
C           ***NEED TO UPDATE PROFILE ENDPOINTS?
               IF (ZI.LE.ZSVP(LP1)) GO TO 10
C           ***YES
               L = L+1
               LP1 = L+1
   10       CWATER(I) = CSVP(L) + (CSVP(LP1)-CSVP(L)) * (ZI-ZSVP(L)) /
     *                  (ZSVP(LP1)-ZSVP(L))
   20    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE INITAL
C
C        (1)  THIS SUBROUTINE INITIALIZES CONSTANTS AND VARIABLES.
C        (2)  VALUES ARE TRANSFERRED TO/FROM MAIN PROGRAM VIA COMMON
C             BLOCK.
C
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      DATA PI/3.141592654/
C
C     *** IF CO NOT SPECIFIED, SET CO TO AVERAGE SPEED IN WATER COLUMN
C     *** (USING MAX DEPTH PROFILE)
          IF(CO.NE.0.0) GO TO 11
          DO 10 I=2,NSVP
             CO=CO+(ZSVP(I)-ZSVP(I-1))*(CSVP(I-1) +
     *             0.5 *(CSVP(I)-CSVP(I-1)))
   10     CONTINUE
          CO = CO/ZSVP(NSVP)
   11     CONTINUE
C
C     *** INITIALIZE RANGE
          RA1 = 0.0
C
C     *** INITIALIZE POINTER THAT POINTS TO BOTTOM PROFILE POINT
          IBOT1 = 0
C
C     *** COMPUTE REFERENCE WAVE NUMBER
          XKO = 2.0*PI*FRQ/CO
C
C     *** COMPUTE REFERENCE WAVELENGTH
          XLAMDA = CO/FRQ
C
C     *** IF DRLVL=0 SET DRLVL EQUAL TO 1/2 REFERENCE WAVELENGTH
          IF ( DRLVL.EQ.0.0 )  DRLVL = 0.5 * XLAMDA
C
C     *** IF DRMAX=0 SET DRMAX EQUAL TO REFERENCE WAVELENGTH
          IF ( DRMAX.EQ.0.0 ) DRMAX = XLAMDA
C
C     *** IF DRLVL GREATER THAN DRMAX SET DRLVL EQUAL TO DRMAX
          IF (DRLVL.GT.DRMAX) DRLVL = DRMAX
C
C     *** COMPUTE ATTENUATION - SACLANT MEMO SM-121 (JENSEN + FERLA)
C     *** MODIFIED AS FOLLOWS:
C     *** IF INPUTTED BETA IS LT 0.0, ALPHA IS COMPUTED IN DB/METER
```

```
C     *** AND USED FOR BETA
          ALPHA=FRQ*FRQ*(.007+(.155*1.7)/(1.7*1.7+FRQ*FRQ*.000001))
     *         *1.0E-09
C
C     *** INITIALIZE POINTER THAT POINTS TO INTERFACE GRID POINT
          IFACE = INT ( BZ(1)/DZ + 0.5 )
C
      RETURN
      END
```

```
      SUBROUTINE MATCON
C
C     THIS SUBROUTINE CALCULATES VARIOUS VARIABLES NEEDED TO COMPUTE
C     TRIDIAGINAL MATRIX ELEMENTS.  VARIABLES BEGINNING WITH XX HAVE
C     NO SPECIAL PHYSICAL SIGNIFICANCE BUT THEY CONTRIBUTE TO
C     COMPUTATIONAL EFFICIENCY.
C
      COMPLEX XN1
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5C00),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
      EYE = CMPLX(0.0,1.0)
C
      XX1 = CMPLX ( 0.0 , +0.25/(DZ*DZ*XKO) )
      XX2 = 2.C * XX1
C     *** COMPUTE COEFFICIENT A IN SEDIMENT LAYER
C        *** FIRST CALCULATE REAL INDEX OF REFRACTION
            XN = CO/C2
C        *** THEN CALCULATE COMPLEX INDEX OF REFRACTION SQUARED
C        ***    (SEE PAGE 2-11 IN TR 6659)
            XN1 = CMPLX ( XN*XN , XN*XN*BETA2/27.287527 )
C        *** CALCLLATE A2
            A2 = 0.5 * EYE * XKO * (XN1-1.0)
      XX3 = 0.5*A2 - XX2
      XX4 = 1.0 + RHO1/RHO2
      XX5 = XX2*(1.0/XX4 - 1.0) + A2/(2.0*(RHC2/RHO1 + 1.0))
      XX6 = XX2/XX4
      XX7 = RHO1/RHO2 * XX6
      XX8 = CMPLX ( 0.0 , -DZ*XKO )
      XX9 = 1.0 / XX8
      XX10 = RHO1 + RHO2
      XX11 = RHO1 - RHO2
      XX12 = 4.0 * XX1
C
```

```
C        *** THIS SECTION PERTAINS TO POINTS  IN WATER COLUMN          MAT00490
            DO 10 I=1,NWMAX                                             MAT00500
C              *** CALCULATE REAL INDEX OF REFRACTION IN WATER          MAT00510
                   XN = CO/CWATER(I)                                    MAT00520
C              *** CALCULATE ATTENUATION AS PER CCMMENTS IN SUBROUTINE  MAT00530
C              *** INITAL                                               MAT00540
                   IF (BETA1.LT.0.0) BETA1 = ALPHA*CWATER(I)/FRQ        MAT00550
C              *** CALCULATE COMPLEX INDEX OF REFRACTION SQUARED        MAT00560
C              ***   (SEE PAGE 2-11 IN TR 6659)                         MAT00570
                   XN1 = CMPLX ( XN*XN , XN*XN*BETA1/27.287527 )        MAT00580
C              *** CALCULATE COEFFICIENT A(I)                           MAT00590
                   A(I) = 0.5 * EYE * XK0 * (XN1-1.0)                   MAT00600
C              *** CALCULATE XX1M                                       MAT00610
                   XX1M(I) = 0.5 * A(I)  -  XX2                         MAT00620
 10         CCNTINUE                                                    MAT00630
      RETURN                                                            MAT00640
      END                                                               MAT00650
```

```
      SUBROUTINE SFIELD(FRQ,CO,ZS,N,DZ,U)
C
C     *** THIS SUBROUTINE IS IDENTICAL TO SUBROUTINE SFIELD AS PER
C     *** NUSC TECHNICAL REPORT 6659.
C
C     ****************************************************************
C     *** GAUSSIAN STARTING FIELD - SEE NORDA TECH NOTE 12 BY H.K.BROCK
C     ****************************************************************
C
C     *** CALLING ROUTINE SUPPLIES:
C         FRQ - FREQUENCY IN HZ
C         CO  - REFERENCE SOUND SPEED - METERS/SEC
C         ZS  - DEPTH OF SOURCE IN METERS.
C         N   - NUMBER OF POINTS IN ARRAY U
C         DZ  - DEPTH INCREMENT - METERS
C     *** SFIELD SUBROUTINE SUPPLIES:
C         U   - COMPLEX STARTING FIELD
C     ****************************************************************
C
      COMPLEX U(1)
      DATA PI/3.1415926535/
C
C     THE FIELD IS DEFINED AS A GAUSSIAN BEAM AT RANGE = 0.
C     LOCAL VARIABLES - GA  GAUSSIAN AMPLITUDE
      XKO=2.0*PI*FRQ/CO
      GW=2.0/XKO
      GA=SQRT(GW)/GW
      DO 10 I=1,N
      ZM=I*DZ
      PR=GAUSS(GA,ZM,ZS,GW)-GAUSS(GA,-ZM,ZS,GW)
      U(I)=CMPLX(PR,0.0)
10    CONTINUE
      RETURN
      END
      FUNCTION GAUSS(GA,Z,GD,GW)
C     INPUT  - GA  GAUSSIAN AMPLITUDE
C     OUTPUT - GAUSS = GA * EXP(-((Z - GD) / GW)**2)
C     TEMPORARY VARIABLE - V
      V=(Z-GD)/GW
      V=-(V*V)
      GAUSS=GA*EXP(V)
      RETURN
      END
```

```
      SUBROUTINE WRITE1
C
C        (1)  THIS SUBROUTINE OUTPUTS UNFORMATTED DATA TO A FILE
C             THAT IS USED BY THE PLOTTING ROUTINE.
C        (2)  THE FILE CORRESPONDS TO UNIT FILE NUMBER:   NOU = 52
C        (3)  THE FILENAME AND FILETYPE FOR THIS FILE ARE:
C                   IFDCUT    PLOTTER
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,CRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      DATA NOU/52/
C
C     *** WRITE MAXIMUM RANGE
          WRITE(NOU,*) RMAX
C
C     *** INITIALIZE RANGE VARIABLE AT WHICH SOLUTION IS TO BE RECORDED
          XWR = RA1+WDR
C
C     *** COMPUTE RECEIVER DEPTH TO NEAREST DZ
          IF ( ZR.LT.DZ ) ZR = DZ
          IWZ = ZR/DZ + 0.5
          ZR = IWZ*DZ
C
C     *** WRITE STARTING VALUE
          WRITE(NOU,*)  RA1, ZR, U(IWZ)
C
      RETURN
      END
```

```
      SUBROUTINE PRINT1
C
C        (1)   THIS SUBROUTINE OUTPUTS FORMATTED DATA TO A FILE
C              WHICH IS READY TO BE SENT TO THE PRINTER.
C        (2)   THE FILE CORRESPONDS TO UNIT FILE NUMBER:  NPOUT = 55
C        (3)   THE FILENAME AND FILETYPE FOR THIS FILE ARE:
C                   IFDOUT     PRINTER
C
      DIMENSION ID(17)
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *           NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *           CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *           R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *           XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      DATA NPOUT/55/
C
C     *** PROMPT USER FOR RUN IDENTIFICATION
          WRITE(6,890)
C     *** READ USER RESPONSE
          READ(5,891) ( ID(I), I=1,16 )
C
C     *** PRINT SELECTED PARAMETERS OF INTEREST
          WRITE(NPOUT,900) (ID(I), I=1,16),   FRQ, ZS, ZR, DZ, CO, XKO,
     *                       XLAMDA, RMAX, DRLVL, DRMAX, WDR, ZABLYR, N
C
C     *** PRINT SOUND SPEED PROFILE IN WATER
          WRITE(NPOUT,904)
          DO 5 I=1,NSVP
             WRITE(NPOUT,905) ZSVP(I), CSVP(I)
    5     CONTINUE
C
C     *** PRINT MORE SELECTED PARAMETERS OF INTEREST
          WRITE(NPOUT,901) ZLYR1, RHO1, BETA1, ZLYR2, RHO2, BETA2, C2
C
C     *** PRINT BOTTOM PROFILE
          WRITE(NPOUT,902)
          NBOTM1 = NBOT-1
          DO 10 I=1,NBOTM1
             WRITE(NPOUT,903) BR(I), BZ(I)
   10     CONTINUE
C
C     *** COMPUTE DEPTH PRINT INCREMENT TO NEAREST DZ
          IPZ = INT ( PDZ/DZ+0.5 )
          IF ( IPZ.EQ.0 ) IPZ = 1
C
C     *** INITIALIZE RANGE VARIABLE AT WHICH SOLUTION IS TO BE PRINTED
          XPR = RA1+PDR
```

```
C                                                                          PRI00490
 890  FORMAT( /,1X,'ENTER RUN IDENTIFICATION:')                            PRI00500
 891  FORMAT(16A4)                                                         PRI00510
 900  FORMAT(///,1X,'IFD PRINTED OUTPUT',//,1X,'RUN I.D. :  ',             PRI00520
     *      16A4,///,7X,'GAUSSIAN STARTING ',                              PRI00530
     *      'FIELD',//,7X,'(FOR FURTHER INFORMATION ON VARIABLES',         PRI00540
     *      ' SEE MAIN IFD PROGRAM LISTING)',//,                           PRI00550
     * 7X,'FREQUENCY                           FRQ    = ',F8.2,' HZ',//  PRI00560
     * 7X,'SOURCE DEPTH                        ZS     = ',F8.2,' M',//   PRI00570
     * 7X,'RECEIVER DEPTH                      ZR     = ',F8.2,' M',//   PRI00580
     * 7X,'DEPTH INCREMENT                     DZ     = ',F8.2,' M',//   PRI00590
     * 7X,'REF SOUND SPEED                     CO     = ',F8.2,' M/S',//PRI00600
     * 7X,'REF WAVE NUMBER                     XKO    = ',F8.2,' 1/M',//PRI00610
     * 7X,'REF WAVELENGTH                      XLAMDA = ',F8.2,' M',//   PRI00620
     * 7X,'MAX RANGE OF SOLUTION               RMAX   = ',F8.1,' M',//   PRI00630
     * 7X,'RANGE STEP ON HORIZONTAL SECTION    DRLVL  = ',F8.2,' M',//   PRI00640
     * 7X,'MAXIMUM RANGE STEP                  DRMAX  = ',F8.2,' M',//   PRI00650
     * 7X,'RECORDED RANGE STEP                 WDR    = ',F8.2,' M',//   PRI00660
     * 7X,'DEPTH OF UPPER EDGE ART ABSORP LYR  ZABLYR = ',F8.2,' M',//   PRI00670
     * 7X,'#VERTICAL POINTS IN GRID             N      = ',I6,//)         PRI00680
 901  FORMAT(/// ' LAYER       MAX DEPTH(M)      DENSITY(G/CM**3)  ',     PRI00690
     *      'ATT(DB/WL)      SOUND SPEED',//                               PRI00700
     *      ' WATER   ',F13.1,5X,F13.2,5X,F13.6,5X,'SEE PROFILE ABOVE'     PRI00710
     *      ,//,' SEDIMENT',F13.1,5X,F13.2,5X,F13.6,4X,F13.2,///)         PRI00720
 902  FORMAT(//,' BOTTOM PROFILE:',//,25X,'RANGE',13X,'DEPTH',/)           PRI00730
 903  FORMAT(20X,F10.1,' M',5X,F11.1,' M',//)                              PRI00740
 904  FORMAT(///,' SOUND SPEED PROFILE IN WATER:',//,20X,'DEPTH',          PRI00750
     *      10X,'SOUND SPEED',//)                                          PRI00760
 905  FORMAT(14X,F10.2,' M', 6X,F10.2,' M/S',/)                            PRI00770
C                                                                          PRI00780
      RETURN                                                               PRI00790
      END                                                                  PRI00800
```

```
      SUBROUTINE NEWSEG
C
C     THIS SUBROUTINE IS CALLED AT THE START OF EACH NEW BOTTOM
C     SEGMENT.  THE SUBROUTINE DOES THE FOLLOWING TASKS FOR EACH
C     BOTTOM SEGMENT:
C         (1)  UPDATES BOTTOM PROFILE POINTERS:  IBOT1 & IBOT2
C         (2)  COMPUTES SLOPE:  THETA
C         (3)  COMPUTES NUMBER OF RANGE STEPS IN SEGMENT:  NSTEP
C         (4)  COMPUTES RANGE STEP:  DR
C         (5)  SETS SLOPE FLAG :  ISLOPE
C                (A)  ISLOPE = 1   -   BOTTOM SLOPES DOWN
C                (B)  ISLOPE = 2   -   BOTTOM LEVEL
C                (C)  ISLOPE = 3   -   BOTTOM SLOPES UP
C                (D)  ISLOPE = 4   -   BOTTOM SLOPES DOWN, MODIFY BOTTOM
C                (E)  ISLOPE = 5   -   BOTTOM SLOPES UP, MODIFY BOTTOM
C         (6)  INIALIZES RANGES:  RA1 & RA2
C         (7)  CHECKS THAT RANGE STEP IS LESS THAN DRMAX
C         (8)  ISSUES ERROR OR WARNING MESSAGES AS APPROPRIATE
C
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *       NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *       CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *       R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *       XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      DATA NPCUT/55/
C
C     *** UPDATE BOTTOM PROFILE POINTER
          IBOT1 = IBOT1 + 1
          IBOT2 = IBOT1 + 1
C     *** GET STARTING AND ENDING RANGES AND DEPTHS FOR THIS SEGMENT
          R1  = BR(IBOT1)
          Z1  = BZ(IBOT1)
          R2  = BR(IBOT2)
          Z2  = BZ(IBOT2)
C         *** ERROR CHECK
              IF (R2.LE.R1) GO TO 100
C         *** PUT Z1 AND Z2 ON NEAREST GRID POINTS
              ITEMP = INT ( Z1/DZ + 0.5 )
              Z1 = DZ * FLOAT(ITEMP)
              ITEMP = INT ( Z2/DZ + 0.5 )
              Z2 = DZ * FLOAT(ITEMP)
C     *** COMPUTE SLOPE
          THETA = ATAN2 (Z2-Z1,R2-R1)
C     *** DOES BOTTOM SLOPE DOWN, LEVEL OR UP?
          IF (THETA.GT.0.0) GO TO 10
          IF (THETA.LT.0.0) GO TO 20
C
```

```
C         *** BOTTOM IS LEVEL
C             *** CETERMINE NUMBER OF RANGE STEPS FOR SEGMENT
                  NSTEP = INT ( (R2-R1)/DRLVL + 0.99999 )
C             *** CETERMINE RANGE STEP
                  DR = (R2-R1) / FLOAT(NSTEP)
C             *** SET ISLOPE
                  ISLOPE = 2
              GO TC 80
C
C         *** BOTTCM SLOPES DOWN
C             *** DETERMINE NUMBER OF RANGE STEPS
 10               NSTEP = INT ( (Z2-Z1+0.05)/DZ )
C             *** DETERMINE RANGE STEP
                  DR = (R2-R1)/FLOAT(NSTEP)
C             *** SET ISLOPE
                  ISLOPE = 1
              GC TO 30
C
C         *** BOTTCM SLOPES UP
C             *** DETERMINE NUMBER OF RANGE STEPS
 20               NSTEP = INT ( (Z1-Z2+0.05)/DZ )
C             *** DETERMINE RANGE STEP
                  DR = (R2-R1)/FLOAT(NSTEP)
C             *** SET ISLOPE
                  ISLOPE = 3
C
 30   CONTINUE
C     *** IS RANGE STEP TOO LARGE?
          IF ( DR.LE.DRMAX ) GO TO 80
C     *** YES, BOTTOM MUST BE MODIFIED
C         *** SET ISLOPE
              ISLCPE = 4
              IF ( THETA.LT.0.0 ) ISLOPE = 5
C         *** DETERMINE NUMBER OF RANGE STEPS REQUIRED TO MOVE UP
C         *** CR DOWN ONE GRID POINT
              NSTEP1 = INT ( DR/DRMAX + 0.99999 )
C         *** DETERMINE RANGE STEP
              DR = DR / FLOAT(NSTEP1)
C         *** REDETERMINE NUMBER OF RANGE STEPS
              NSTEP = NSTEP * NSTEP1
C         *** COMPUTE SLOPE OF SLOPING SECTION
              THETA = ATAN2(DZ,DR)
C         *** COMPUTE LOCATION OF NEXT LEVEL SECTION FOLLOWING A
C         *** SLOPING SECTION
              NXLFS = NSTEP1/2 + 2
C         *** INDICATE TO USER THAT BOTTOM HAS BEEN MODIFIED
              TEMP = 0.5 * DZ
              WRITE(6,903) R1,R2,TEMP
```

```
            WRITE(NPOUT,903) R1,R2,TEMP
C
 80    CONTINUE
C      *** INITIALIZE RA1 & RA2
           RA1 = R1
           RA2 = RA1+DR
C
C      *** INDICATE TO USER HOW FAR SOLUTION FIELD HAS PROGRESSED
           WRITE(6,902) R1
C
C      *** IF RANGE STEP GREATER THAN 1 (?) WAVELENGTH WRITE WARNING
           IF ( DR.LE.XLAMDA ) GO TO 90
              WRITE(6,901) R1, R2, DR, XLAMDA
              WRITE(NPOUT,901) R1, R2, DR, XLAMDA
C
 90    RETURN
C
C      *** ERROR EXIT
 100       WRITE(6,900) IBOT2, IBOT1
           WRITE(NPOUT,900) IBOT2, IBOT1
C
           STOP
C
 900   FORMAT(//,1X,'ERROR:  THE RANGE AT BOTTOM PROFILE POINT NUMBER ',
      *       I2,' IS LESS'/,9X,'THAN THE RANGE AT BOTTOM PROFILE POINT'
      *       ,' NUMBER ',I2,'.',//,1X,'EXECUTION TERMINATED.'////)
 901   FORMAT(/,
      * ' WARNING:  THE HORIZONTAL RANGE STEP BETWEEN RANGE R  =',F8.1,/
      * '           AND RANGE R =',F8.1,' (METERS) IS',F5.1,' METERS.',/,
      * '           THE REFERENCE WAVELENGTH IS',F5.1,' METERS.')
 902   FORMAT(/,' THE PROGRAM HAS REACHED RANGE R =',F8.1,' METERS.')
 903   FORMAT(/,' NOTE:  THE BOTTOM BETWEEN RANGE',F8.1,' AND RANGE',
      *       F8.1,/,8X,'HAS BEEN MODIFIED BECAUSE OF ITS VERY SMALL',
      *       ' SLOPE.',/,8X,'THE DIFFERENCE BETWEEN THE MODIFIED ',
      *       'BOTTOM AND YOUR',/,8X,'INPUT BOTTOM IS NEVER GREATER ',
      *       'THAN',F5.2,' METERS.'/)
C
       END
```

```
      SUBROUTINE NEWMAT
C
C
C     THIS SUBROUTINE IS CALLED AT THE START OF EACH NEW BOTTOM
C     SEGMENT.  THE SUBROUTINE DOES THE FOLLOWING TASKS:
C       (1)   COMPUTES TRIDIAGONAL MATRIX ELEMENTS FOR MATRIX Y
C             (A)  MATRIX Y IS AT RANGE WHERE FIELD IS KNOWN: RANGE=RA1
C             (B)  Y MATRIX ON RHS OF EQUATION
C       (2)   USES Y MATRIX ELEMENTS AND KNOWN FIELD AT RANGE RA1 TO
C             COMPUTE RHS COLUMN VECTOR C(I,4).    (SEE TRIDG AND TEXT)
C       (3)   COMPUTES TRIDIAGONAL MATRIX ELEMENTS FOR MATRIX X
C             (A)  X MATRIX ON LHS OF EQUATION
C             (B)  X MATRIX VALUES STORED IN C(I,1), C(I,2) AND C(I,3)
C                  (SEE TRIDG AND TEXT REFERENCED IN TRIDG)
C             (C)  X MATRIX AT RANGE:  RA2 = RA1 + DR
C
C     TRIDIAGONAL MATRIX ELEMENTS ARE REPRESENTED BY VARIABLES
C     BEGINNING WITH EITHER X OR Y.  FOR VARIABLES BEGINNING WITH X OR
C     Y, THE LETTERS IN THE VARIABLE NAME HAVE THE FOLLOWING
C     SIGNIFICANCE:
C                       X  -  ELEMENT IN X MATRIX
C                       Y  -  ELEMENT IN Y MATRIX
C                       M  -  ELEMENT ON MAIN DIAGONAL
C                       L  -  ELEMENT TO LEFT OF MAIN DIAGONAL
C                       R  -  ELEMENT TO RIGHT OF MAIN DIAGONAL
C                       W  -  ELEMENT IN ROW IN WATER
C                       I  -  ELEMENT IN ROW AT INTERFACE
C                       S  -  ELEMENT IN ROW IN SEDIMENT
C                       V  -  INTERFACE ELEMENT ON LEVEL INTERFACE
C                       Z  -  INTERFACE ELEMENT ON SLOPING INTERFACE
C
      COMPLEX DELTA,DELIN,BEDA1,GAMMA1,BEDA2,GAMMA2,ZZ1,ZZ2,ZZ3,ZZ4
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,CRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
```

```
C                                                                       NEW00490
C     *** COMPUTE MAIN DIAGONAL ELEMENT, Y MATRIX, IN SEDIMENT          NEW00500
          YMS = 1.0 + DR*XX3                                            NEW00510
C     *** COMPUTE OFF-DIAGONAL ELEMENTS, Y MATRIX, IN WATER & SEDIMENT  NEW00520
          YLRWS = DR*XX1                                                NEW00530
C     *** COMPUTE MAIN DIAGONAL ELEMENT, X MATRIX, IN SEDIMENT          NEW00540
          XMS = 2.0 - YMS                                               NEW00550
C     *** COMPUTE OFF-DIAGONAL ELEMENTS, X MATRIX, IN WATER & SEDIMENT  NEW00560
          XLRWS = -YLRWS                                                NEW00570
C     *** COMPUTE FIRST ELEMENT IN RHS COLUMN VECTOR                    NEW00580
          YMW(1) = 1.0 + DR*XX1M(1)                                     NEW00590
          C(1,4) = U(1)*YMW(1) + U(2)*YLRWS                             NEW00600
C     *** COMPUTE TWO ELEMENTS IN FIRST ROW ON LHS                      NEW00610
          C(1,2) = 2.0 - YMW(1)                                         NEW00620
          C(1,3) = XLRWS                                                NEW00630
C     *** COMPUTE REMAINING ELEMENTS ON BOTH RHS & LHS FOR ROWS IN WATERNEW00640
          IFACEM = IFACE - 1                                            NEW00650
          DO 10 I=2,IFACEM                                              NEW00660
C               *** FIRST WORK WITH RHS                                 NEW00670
                    YMW(I) = 1.0 + DR*XX1M(I)                           NEW00680
                    C(I,4) = U(I)*YMW(I) + (U(I-1)+U(I+1))*YLRWS        NEW00690
C               *** WORK WITH LHS                                       NEW00700
                    C(I,1) = XLRWS                                      NEW00710
                    C(I,2) = 2.0 - YMW(I)                               NEW00720
                    C(I,3) = XLRWS                                      NEW00730
   10     CONTINUE                                                      NEW00740
C     *** COMPUTE LHS & RHS ELEMENTS IN SEDIMENT                        NEW00750
          IFACEP = IFACE + 1                                            NEW00760
          NM1 = N - 1                                                   NEW00770
          DO 20 I=IFACEP,NM1                                            NEW00780
C               *** RHS                                                 NEW00790
                    C(I,4) = U(I)*YMS + (U(I-1)+U(I+1))*YLRWS           NEW00800
C               *** LHS                                                 NEW00810
                    C(I,1) = XLRWS                                      NEW00820
                    C(I,2) = XMS                                        NEW00830
                    C(I,3) = XLRWS                                      NEW00840
   20     CONTINUE                                                      NEW00850
C                                                                       NEW00860
C     *** IF ENTIRE SEGMENT LEVEL GO TO 50                              NEW00870
          IF ( ISLOPE.EQ.2 ) GO TO 50                                   NEW00880
C                                                                       NEW00890
C     *** INTERFACE SLOPES EITHER UP OR DOWN                            NEW00900
C     *** CALCULATE CONSTANTS FOR COMPUTING MATRIX ELEMENTS             NEW00910
          THETA = ABS (THETA)                                           NEW00920
          SINE = SIN (THETA)                                            NEW00930
          COSE = COS (THETA)                                            NEW00940
          DELTA = XX10 + XX11*(XX8+XX9)*SINE*COSE                       NEW00950
          DELIN = 1.0 / DELTA                                           NEW00960
```

```
      BEDA1 = DELIN * XX12 * RHO1
      GAMMA1 = DELIN * (XX12*(RHO2*COSE*COSE+RHO1*SINE*SINE) +
     *         XX11*SINE*COSE/DZ)
      BEDA2 = DELIN  * (XX12*(RHO1*COSE*COSE+RHO2*SINE*SINE) +
     *         XX11*SINE*COSE/DZ)
      GAMMA2 = DELIN * XX12 * RHO2
      ZZ1 = DELIN * (RHO1*SINE*SINE + RHO2*COSE*COSE +
     *      XX8*SINE*COSE*XX11)
      ZZ2 = DELIN * (RHO1*A2 - (COSE-XX8*SINE)*XX9*XX11*EYE*XKO*
     *      SINE)
      ZZ3 = DELIN * RHO2
      ZZ4 = DELIN * (  A2 * ( RHO1*COSE*COSE + RHO2*SINE*SINE
     *      + XX8*SINE*COSE*XX11 )  -
     *      ( COSE + XX8*SINE )*XX9*XX11*EYE*XKO*SINE   )
      ZZ5 = 0.5*DR*ZZ1
      ZZ6 = 1.0 + 0.5*DR*(ZZ2-BEDA1-GAMMA1)
      ZZ7 = -0.5*DR*ZZ3
      ZZ8 = 1.0 - 0.5*DR*(ZZ4-BEDA2-GAMMA2)
      ZZ9 = 2.0 - ZZ8
      ZZ10 = 2.0 - ZZ6
C
C     *** IF BOTTOM SLOPES UP, GO TO 40
          IF ( ISLOPE.EQ.3 .OR. ISLOPE.EQ.5 ) GO TO 40
C     *** BOTTOM SLOPES DOWN
          IFACE2 = IFACEP
C         *** COMPUTE OFF-DIAGONAL, Y MATRIX ELEMENTS ON INTERFACE
              YLI = 0.5 * DR * GAMMA1
              YRI = 0.5 * DR * BEDA1
C         *** COMPUTE MAIN DIAGONAL, Y MATRIX ELEMENT ON INTERFACE
              YMI = A(IFACE) * ZZ5  +  ZZ6
C         *** COMPUTE INTERFACE ELEMENT IN RHS COLUMN VECTOR
              C(IFACE,4) = U(IFACEM)*YLI + U(IFACE)*YMI +
     *                     U(IFACEP)*YRI
C         *** COMPUTE X MATRIX ELEMENTS ON INTERFACE
              XLI = -0.5 * DR * GAMMA2
              XMI = A(IFACE2) * ZZ7  +  ZZ8
              XRI = -0.5 * DR * BEDA2
C             *** IF MODIFIED BOTTOM THEN NO NEED TO ADJUST LHS
                  IF ( ISLOPE.EQ.4 ) GO TO 45
              C(IFACE2,1) = XLI
              C(IFACE2,2) = XMI
              C(IFACE2,3) = XRI
C         *** COMPUTE X MATRIX ELEMENTS ONE ROW ABOVE INTERFACE
              C(IFACE,1) = XLRWS
              C(IFACE,2) = 1.0 - DR * XX1M(IFACE)
              C(IFACE,3) = XLRWS
          GO TO 60
C
```

```
C       *** BOTTOM SLOPES UP                                              NEW01450
 40         IFACE2 = IFACEM                                               NEW01460
C           *** COMPUTE OFF-DIAGONAL, Y MATRIX ELEMENTS ON INTERFACE      NEW01470
                YLI = 0.5 * DR * GAMMA2                                   NEW01480
                YRI = 0.5 * DR * BEDA2                                    NEW01490
C           *** COMPUTE MAIN DIAGONAL, Y MATRIX ELEMENT ON INTERFACE      NEW01500
                YMI = - A(IFACE) * ZZ7  +  ZZ9                            NEW01510
C           *** COMPUTE INTERFACE ELEMENT IN RHS COLUMN VECTOR            NEW01520
                C(IFACE,4) = U(IFACEM)*YLI + U(IFACE)*YMI +               NEW01530
     *                       U(IFACEP)*YRI                                NEW01540
C           *** COMPUTE X MATRIX ELEMENTS ON INTERFACE                    NEW01550
                XLI = -0.5 * DR * GAMMA1                                  NEW01560
                XMI = - A(IFACE2) * ZZ5  +  ZZ10                          NEW01570
                XRI = -0.5 * DR * BEDA1                                   NEW01580
C               *** IF MODIFIED BOTTOM THEN NO NEED TO ADJUST LHS         NEW01590
                   IF ( ISLOPE.EQ.5 ) GO TO 45                            NEW01600
                C(IFACE2,1) = XLI                                         NEW01610
                C(IFACE2,2) = XMI                                         NEW01620
                C(IFACE2,3) = XRI                                         NEW01630
C           *** COMPUTE X MATRIX ELEMENTS ONE ROW BELOW INTERFACE         NEW01640
                C(IFACE,1) = XLRWS                                        NEW01650
                C(IFACE,2) = XMS                                          NEW01660
                C(IFACE,3) = XLRWS                                        NEW01670
            GO TO 60                                                      NEW01680
C                                                                         NEW01690
C       *** SAVE INTERFACE VALUES ON SLOPING SECTION                      NEW01700
 45         YLIZ = YLI                                                    NEW01710
            YRIZ = YRI                                                    NEW01720
            XLIZ = XLI                                                    NEW01730
            XRIZ = XRI                                                    NEW01740
C                                                                         NEW01750
C       *** SEGMENT LEVEL, COMPUTE MATRIX ELEMENTS ON INTERFACE           NEW01760
 50         IFACE2 = IFACE                                                NEW01770
            YLI = DR * XX6                                                NEW01780
            YMI = 1.0 + DR * ( XX1M(IFACE)/XX4 + XX5 )                    NEW01790
            YRI = DR * XX7                                                NEW01800
            C(IFACE,4) = U(IFACEM)*YLI + U(IFACE)*YMI + U(IFACEP)*YRI     NEW01810
            C(IFACE,1) = -YLI                                             NEW01820
            C(IFACE,2) = 2.0 - YMI                                        NEW01830
            C(IFACE,3) = -YRI                                             NEW01840
C           *** SAVE INTERFACE VALUES ON LEVEL SECTION                    NEW01850
                YLIV = YLI                                                NEW01860
                YRIV = YRI                                                NEW01870
C                                                                         NEW01880
 60     CONTINUE                                                          NEW01890
C                                                                         NEW01900
C       *** COMPUTE ARTIFICIAL ATTENUATION MATRIX                         NEW01910
C       *** SEE AESC PE MODEL BY BROCK  -  NORDA TECH NOTE 12  -  JAN 78  NEW01920
```

```
C     *** CALCULATE GRID POINT AT TOP OF ART ATTENUATION LAYER
          IA = INT ( ZABLYR/DZ + 0.01 )
C     *** CALCULATE NUMBER OF GRID POINTS IN ART ATTENUATION LAYER
          NA = N - IA
C     *** CALCULATE ATTENUATION MATRIX
          DO 70 I=1,NA
             TEMP = 3.0 * (I-NA)  /  NA
             ATT(I) = EXP(-0.01*DR*EXP(-(TEMP*TEMP)))
 70       CONTINUE
C
C     *** SOLVE FOR SOLUTION FIELD AT RANGE RA2
          CALL TRIDG (C,U,N,CR,CTWO)
C
C     *** APPLY ARTIFICAL ATTENUATION
          CALL ATTENU (U,ATT,IA,NA)
C
      RA2P = RA2 + 0.5
C     *** TIME TO WRITE?
          IF ( RA2P.GE.XWR )  CALL WRITE2
C     *** TIME TO PRINT?
          IF ( RA2P.GE.XPR )  CALL PRINT2
C
C     *** UPDATE INTERFACE POINTER
          IFACE = IFACE2
C
      RETURN
      END
```

```
      SUBROUTINE TRIDG (C,U,N,CR,CTWO)
C
C     *** THIS SUBROUTINE SOLVES A SET OF  N - 1  (NM1)   LINEAR
C     *** SIMULTANEOUS EQUATIONS HAVING A TRIDIAGONAL COEFFICIENT
C     *** MATRIX.  MATRIX ELEMENTS IN THE LOWER DIAGONAL, MAIN DIAGONAL
C     *** AND UPPER DIAGONAL ARE STORED IN C(I,1), C(I,2) AND C(I,3)
C     *** RESPECTIVELY.  THE RHS COLUMN VECTOR IS STORED IN C(I,4).
C     *** THE SOLUTION FIELD IS STORED IN U(I).
C     ***     (1)  THE INDEX I REFERS TO ROW NUMBER.
C     ***     (2)  WE NEED ONLY SOLVE AN  NM1 X NM1  SYSTEM (RATHER THAN
C     ***          AN  N X N  SYSTEM) BECAUSE U(N) IS KNOWN:  U(N)=0.0
C     ***     (3)  THE SUBROUTINE IS A MODIFIED VERSION OF SUBROUTINE
C     ***          TRIDG FROM:
C     ***              "APPLIED NUMERICAL ANALYSIS"   ( SECOND EDITION )
C     ***              BY:  CURTIS F. GERALD
C     ***              PUBLISHED BY ADDISON-WESLEY PUBLISING CO., 1980
C     ***     (4)  THE MAIN MODIFICATIONS TO THE ROUTINE IN THE TEXT
C     ***          INVOLVED:
C     ***         (A)  INTRODUCING ARRAYS CTWO AND CR TO PRESERVE THE
C     ***              ORIGINAL VALUES IN C(I,2) AND TO MAKE THE ROUTINE
C     ***              MORE EFFICIENT.  THIS RESULTS IN A CONSIDERABLE
C     ***              SAVINGS IN EXECUTION TIME FOR THE CASE OF A
C     ***              HORIZONTAL BOTTOM.   (SEE SUBROUTINE TRIDL)
C     ***         (B)  MODIFYING THE ROUTINE TO SOLVE AN  NM1 X NM1
C     ***              SYSTEM.
C     ***      (5)  SEE PAGES 129 AND 133 IN THE TEXT FOR FURTHER INFO.
C
      COMPLEX C(5000,4), U(5000), CR(5000), CTWO(5000)
C
      NM1 = N - 1
      NM2 = N - 2
      CTWO(1) = C(1,2)
      DO 10  I=2,NM1
         CR(I) = C(I,1) / CTWO(I-1)
         CTWO(I) = C(I,2) - CR(I) * C(I-1,3)
         C(I,4)  = C(I,4) - CR(I) * C(I-1,4)
   10 CONTINUE
C
      U(N) = 0.0
C
C     *** NOW PERFORM BACK SUBSTITUTION
C
      U(NM1) = C(NM1,4) / CTWO(NM1)
      DO 20  I=1,NM2
         M = NM1 - I
         U(M) = ( C(M,4)-C(M,3)*U(M+1) ) / CTWO(M)
   20 CONTINUE
C
```

```
      RETURN                                                    TRI00490
      END                                                       TRI00500
```

```
      SUBROUTINE TRIDL (C,U,N,CR,CTWO)
C
C     *** THIS SUBROUTINE IS A MODIFIED VERSION OF SUBROUTINE TRIDG
C     *** FROM THE IFD PROGRAM.  SUBROUTINE TRIDG IS IN TURN A MODIFIED
C     *** VERSION OF TRIDG AS PER THE REFERENCE BELOW.
C     *** THE SUBROUTINE SOLVES A SET OF  N - 1  (NM1)  LINEAR
C     *** SIMULTANEOUS EQUATIONS HAVING A TRIDIAGONAL COEFFICIENT
C     *** MATRIX.  MATRIX ELEMENTS IN THE LOWER DIAGONAL, MAIN DIAGONAL
C     *** AND UPPER DIAGONAL ARE STORED IN C(I,1), C(I,2) AND C(I,3)
C     *** RESPECTIVELY.  THE RHS COLUMN VECTOR IS STORED IN C(I,4).
C     *** THE SOLUTION FIELD IS STORED IN U(I).
C     ***     (1)  THE INDEX I REFERS TO ROW NUMBER.
C     ***     (2)  WE NEED ONLY SOLVE AN  NM1 X NM1  SYSTEM (RATHER THAN
C     ***          AN  N X N  SYSTEM) BECAUSE U(N) IS KNOWN:  U(N)=0.0
C     ***     (3)  THE SUBROUTINE IS A MODIFIED VERSION OF IFD SUB-
C     ***          ROUTINE TRIDG WHICH IN TURN IS A MODIFIED VERSION
C     ***          OF SUBROUTINE TRIDG AS PER:
C     ***              "APPLIED NUMERICAL ANALYSIS"   ( SECOND EDITION )
C     ***               BY:  CURTIS F. GERALD
C     ***               PUBLISHED BY ADDISON-WESLEY PUBLISING CO., 1980
C     ***     (4)  THE ONLY MODIFICATION TO IFD SUBROUTINE TRIDG IS
C     ***          THAT TRIDL DOES NOT RECALCULATE CTWO AND CR, BUT
C     ***          BUT RATHER TRIDL USES THE ARRAY VALUES CALCULATED
C     ***          BY TRIDG.  THIS RESULTS IN A CONSIDERABLE SAVINGS
C     ***          IN EXECUTION TIME FOR THE CASE OF A HORIZONTAL
C     ***          BOTTOM.
C
      COMPLEX C(5000,4), U(5000), CR(5000), CTWO(5000)
C
      NM1 = N - 1
      NM2 = N - 2
      DO 10 I=2,NM1
         C(I,4) = C(I,4) - CR(I) * C(I-1,4)
   10 CONTINUE
C
      U(N) = 0.0
C
C     *** NOW PERFORM BACK SUBSTITUTION
C
      U(NM1) = C(NM1,4) / CTWO(NM1)
      DO 20 I=1,NM2
         M = NM1 - I
         U(M) = ( C(M,4)-C(M,3)*U(M+1) ) / CTWO(M)
   20 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE DOWN
C
C     THIS SUBROUTINE UPDATES THE RHS & LHS OF THE EQUATION AND
C     SOLVES FOR THE SOLUTION FIELD AT RA2.
C         (1)   THE RHS COLUMN VECTOR VALUES ARE STORED IN  C(I,4)
C         (2)   THE INTERFACE AT RANGE RA1 IS AT GRIDPOINT  IFACE
C         (3)   THE INTERFACE AT RANGE RA2 IS AT GRIDPOINT  IFACE2
C                 ( WHERE IFACE2 = IFACE + 1 )
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),C0,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,CRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C     *** UPDATE IFACE2
          IFACE2 = IFACE + 1
C     *** UPDATE Y MATRIX, MAIN DIAGONAL, INTERFACE ELEMENT
          YMI = A(IFACE) * ZZ5  +  ZZ6
C     *** UPDATE Y MATRIX, MAIN DIAGONAL, WATER ELEMENT, ONE ROW
C     *** ABOVE INTERFACE
          YMW(IFACE-1) = 1.0 + DR * XX1M(IFACE-1)
C
C     *** UPDATE RHS
          CALL RHS
C     *** UPDATE LHS
C        *** UPDATE X MATRIX ELEMENTS ONE ROW ABOVE INTERFACE
              C(IFACE,1) = XLRWS
              C(IFACE,2) = 1.0 - DR*XX1M(IFACE)
              C(IFACE,3) = XLRWS
C        *** UPDATE X MATRIX ELEMENTS ON INTERFACE
              C(IFACE2,1) = XLI
              C(IFACE2,2) = A(IFACE2) * ZZ7  +  ZZ8
              C(IFACE2,3) = XRI
C
C     *** SOLVE THE TRIDIAGONAL SYSTEM
          CALL TRIDG (C,U,N,CR,CTWO)
```

```
C       *** UPDATE IFACE                                                DOW00490
           IFACE = IFACE2                                               DOW00500
        RETURN                                                          DOW00510
        END                                                             DOW00520
```

```
      SUBROUTINE COWN
C
C     THIS SUBROUTINE UPDATES THE RHS & LHS OF THE EQUATION AND
C     SOLVES FOR THE SOLUTION FIELD AT RA2.
C        (1)   THE RHS COLUMN VECTOR VALUES ARE STORED IN C(I,4)
C        (2)   THE INTERFACE AT RANGE RA1 IS AT GRIDPOINT IFACE
C        (3)   THE INTERFACE AT RANGE RA2 IS AT GRIDPOINT IFACE2
C                 ( WHERE IFACE2 = IFACE + 1 )
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,CRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C     *** UPDATE IFACE2
          IFACE2 = IFACE + 1
C     *** UPDATE Y MATRIX, MAIN DIAGONAL, INTERFACE ELEMENT
          YMI = A(IFACE) * ZZ5  +  ZZ6
C     *** UPDATE Y MATRIX, MAIN DIAGONAL, WATER ELEMENT, ONE ROW
C     *** ABOVE INTERFACE
          YMW(IFACE-1) = 1.0 + DR * XX1M(IFACE-1)
C
C     *** UPDATE RHS
          CALL RHS
C     *** UPDATE LHS
C        *** UPDATE X MATRIX ELEMENTS ONE ROW ABOVE INTERFACE
             C(IFACE,1) = XLRWS
             C(IFACE,2) = 1.0 - DR*XX1M(IFACE)
             C(IFACE,3) = XLRWS
C        *** UPDATE X MATRIX ELEMENTS ON INTERFACE
             C(IFACE2,1) = XLI
             C(IFACE2,2) = A(IFACE2) * ZZ7  +  ZZ8
             C(IFACE2,3) = XRI
C
C     *** SOLVE THE TRIDIAGONAL SYSTEM
          CALL TRIDG (C,U,N,CR,CTWO)
```

```
C        *** UPCATE IFACE                                                DOW0C490
            IFACE = IFACE2                                               DOW00500
         RETURN                                                          DOW0C510
         END                                                             DOW00520
```

```
      SUBROUTINE LP
C
C     THIS SUBROUTINE UPDATES THE RHS & LHS OF THE EQUATION AND
C     SOLVES FOR THE SOLUTION FIELD AT RA2.
C        (1)  THE RHS COLUMN VECTOR VALUES ARE STORED IN C(I,4)
C        (2)  THE INTERFACE AT RANGE RA1 IS AT GRIDPOINT IFACE
C        (3)  THE INTERFACE AT RANGE RA2 IS AT GRIDPOINT IFACE2
C                ( WHERE IFACE2 = IFACE - 1 )
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,FA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C     *** UPDATE IFACE2
          IFACE2 = IFACE - 1
C     *** UPDATE Y MATRIX, MAIN DIAGONAL, INTERFACE ELEMENT
          YMI = -A(IFACE) * ZZ7  +  ZZ9
C     *** UPDATE RHS
          CALL RHS
C     *** UPDATE LHS
C        *** UPDATE X MATRIX ELEMENTS ONE ROW BELOW INTERFACE
             C(IFACE,1) = XLRWS
             C(IFACE,2) = XMS
             C(IFACE,3) = XLRWS
C        *** UPDATE X MATRIX ELEMENTS ON INTERFACE
             C(IFACE2,1) = XLI
             C(IFACE2,2) = -A(IFACE2) * ZZ5  +  ZZ10
             C(IFACE2,3) = XRI
C     *** SOLVE THE TRDIAGONAL SYSTEM
          CALL TRIDG (C,U,N,CR,CTWO)
C     *** UPDATE IFACE
          IFACE = IFACE2
C
      RETURN
      END
```

```
      SUBROUTINE SSLOPE                                                 SSL00010
C                                                                       SSL00020
C     *** THIS SUBROUTINE IS CALLED TO ADVANCE THE SOLUTION FIELD       SSL00030
C     *** FOR THE CASE OF A MODIFIED BOTTCM.                            SSL00040
C     ***    (1)  THIS CASE OCCURS WHEN THE BOTTOM SLOPE IS TOO SMALL   SSL00050
C     ***         FOR THE MAXIMUM RANGE STEP.                           SSL00060
C     ***    (2)  THIS SUBROUTINE WORKS FOR BOTH A DOWNSLOPE AND        SSL00070
C     ***         UPSLOPE MODIFIED BOTTOM.                              SSL00080
C     ***    (3)  THE SUBROUTINE DETERMINES WHICH OF THE FOLLOWING      SSL00090
C     ***         THREE TYPES OF BOTTOM SECTIONS NEEDS TO BE CONSIDERED:SSL00100
C     ***           (A) LEVEL SECTION FOLLOWS LEVEL SECTION             SSL00110
C     ***           (B) LEVEL SECTION FOLLOWS SLOPING SECTION           SSL00120
C     ***           (C) SLOPING SECTION.                                SSL00130
C     ***    (4)  AFTER DETERMINING WHICH OF THE THREE TYPES OF BOTTOM  SSL00140
C     ***         SECTIONS IS APPROPRIATE, THE SUBROUTINE MAKES MATRIX  SSL00150
C     ***         ELEMENT CHANGES AS REQUIRED AND CALLS ON OTHER        SSL00160
C     ***         SUBROUTINES TO ADVANCE THE SOLUTION.                  SSL00170
C                                                                       SSL00180
      COMPLEX A,A2,C,CR,CTWO,EYE,                                       SSL00190
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,                          SSL00200
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,                SSL00210
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,            SSL00220
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10                                SSL00230
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,    SSL00240
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS                             SSL00250
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,     SSL00260
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,  SSL00270
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,  SSL00280
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR           SSL00290
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),          SSL00300
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,                      SSL00310
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),          SSL00320
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,      SSL00330
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10                          SSL00340
C                                                                       SSL00350
C     *** IS THIS A LEVEL SECTION FOLLOWING A SLOPING SECTION?          SSL00360
          IF (ISTEP.EQ.NXLFS)  GO TO 20                                 SSL00370
C     *** NO                                                            SSL00380
C                                                                       SSL00390
C     *** IS THIS A SLOPING SECTION?                                    SSL00400
          ITEST = NXLFS - ISTEP                                         SSL00410
          IF (ITEST.EQ.1) GO TO 30                                      SSL00420
C     *** NO                                                            SSL00430
C                                                                       SSL00440
C     *** LEVEL SECTION FOLLOWS A LEVEL SECTION                         SSL00450
          CALL LEVEL                                                    SSL00460
          GO TO 50                                                      SSL00470
C                                                                       SSL00480
```

```
C      *** LEVEL SECTICN FOLLOWS A SLCPING SECTION
C          *** UPDATE NXLFS
 20            NXLFS = NXLFS + NSTEP1
C          *** IF LAST SECTION SLOPED DOWN, UPDATE Y MATRIX ELEMENT,
C          *** MAIN DIAGONAL, IN WATER, ONE ROW ABOVE INTERFACE
                   IF ( ISLOPE.EQ.5 ) GO TC 25
                   YMW(IFACE-1) = 1.0 + DR * XX1M(IFACE-1)
C          *** UPDATE RHS INTERFACE ELEMENTS
 25            YLI  = YLIV
               YMI  = 1.0 + DR * ( XX1M(IFACE)/XX4 + XX5 )
               YRI  = YRIV
C          *** UPDATE LHS
               C(IFACE,1)  = -YLI
               C(IFACE,2)  = 2.0 - YMI
               C(IFACE,3)  = -YRI
C          *** SOLVE SYSTEM
               CALL RHS
               CALL TRIDG(C,U,N,CR,CTWO)
           GO TC 5C
C
C      *** SLOPING SECTION
C          *** UPDATE INTERFACE ELEMENTS
 30            YLI  = YLIZ
               YRI  = YRIZ
               XLI  = XLIZ
               XRI  = XRIZ
C          *** SOLVE SYSTEM AS APPROPRIATE
               IF (ISLOPE.EQ.4) CALL DOWN
               IF (ISLOPE.EQ.5) CALL UP
C
 50    CONTINUE
       RETURN
       END
```

```
      SUBROUTINE RHS
C
C     THIS SUBROUTINE MULTIPLIES TRIDIAGONAL MATRIX Y TIMES SOLUTION
C     FIELD U TO CBTAIN AN UPDATED RHS.
C        (1)  THE RHS COLUMN VECTOR VALUES ARE STCRED IN C(I,4).
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,CRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5C00),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C     *** UPDATE IFACEM & IFACEP
          IFACEM = IFACE - 1
          IFACEP = IFACE + 1
C     *** UPDATE RHS
          C(1,4) = U(1) * YMW(1)  +  U(2) * YLRWS
          DO 10 I=2,IFACEM
             C(I,4) = U(I) * YMW(I)  +  (U(I-1)+U(I+1)) * YLRWS
 10       CONTINUE
          C(IFACE,4) = U(IFACEM)*YLI + U(IFACE)*YMI + U(IFACEP)*YRI
          DO 20 I=IFACEP,NM1
             C(I,4) = U(I)*YMS + (U(I-1)+U(I+1))*YLRWS
 20       CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE LEVEL
C
C     THIS SUBROUTINE UPDATES THE RHS OF THE EQUATION AND SOLVES
C     FOR THE SOLUTION FIELD AT RANGE RA2.
C        (1)  THE RHS COLUMN VECTOR VALUES ARE STORED IN C(I,4).
C        (2)  FOR THE LEVEL INTERFACE THE LHS TRIDIAGINAL MATRIX
C             ELEMENTS NEED NOT BE UPDATED.
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C     *** UPDATE RHS
         CALL RHS
C
C     *** SOLVE THE TRIDIAGONAL SYSTEM
         CALL TRIDL (C,U,N,CR,CTWO)
C
      RETURN
      END
```

```
      SUBROUTINE PRINT2
C
C        (1)   THIS SUBROUTINE IS EFFECTIVELY THE CONTINUATION OF
C              SUBROUTINE PRINT1.
C        (2)   THE FILE CREATED CORRESPONDS TO UNIT FILE NUMBER:
C              NPOLT = 55.
C        (3)   THE FILENAME AND FILETYPE FOR THIS FILE ARE:
C                     IFDOUT     PRINTER
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      DATA NPOUT/55/
C
C     *** PRINT RANGE
          WRITE(NPOUT,900) RA2
C
C     *** COMPUTE AND PRINT PROPAGATION LOSS AT EACH IPZ'TH DEPTH
          WRITE(NPOUT,901)
          DO 20 I=IPZ,N,IPZ
             ZI = I*DZ
             PL = CABS(U(I))
             IF ( PL.LE.0.0 ) GO TO 10
             PL = -20.0*ALOG10(PL) + 10.0*ALOG10(RA2)
             GO TO 15
 10          PL = 999.9
 15          WRITE(NPOUT,902)  ZI, PL, U(I)
 20       CONTINUE
C
C     *** DETERMINE NEXT RANGE AT WHICH TO PRINT SOLUTION
          XPR = XPR+PDR
C
 900  FORMAT(///,5X,'RANGE =',F8.1,' M',/)
 901  FORMAT(15X,'DEPTH',6X,'LOSS(DB)',14X,'U(I)',/)
 902  FORMAT(10X,F10.2,3X,F10.2,3X,'(',E12.5,2X,E12.5,')')
```

```
C                                                                 PRI00490
      RETURN                                                      PRI00500
      END                                                         PRI00510
```

```
      SUBROUTINE WRITE2
C
C        (1)  THIS SUBROUTINE IS EFFECTIVELY THE CONTINUATION OF
C             SUBROUTINE WRITE1
C        (2)  THE SUBROUTINE WRITES RANGE, RECEIVER DEPTH AND U(I)
C             WHEN CALLED.  IT THEN UPDATES THE NEXT WRITE RANGE (XWR).
C        (3)  THE FILE WRITTEN INTO CORRESPONDS TO UNIT FILE NUMBER:
C             NOU = 52.
C        (4)  THE FILENAME AND FILETYPE FOR THIS FILE ARE:
C                  IFDOUT   PLOTTER
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      DATA NOU/52/
C
C     *** WRITE RANGE, DEPTH AND U(I)
          WRITE(NOU,*)  RA2, ZR, U(IWZ)
C
C     *** DETERMINE NEXT RANGE AT WHICH TO WRITE SOLUTION
          XWR = XWR+WDR
C
      RETURN
```

```
      SUBROUTINE ATTENU(U,ATT,IA,NA)
C
C     THIS SUBROUTINE APPLIES ARTIFICIAL ATTENUATION TO THE BOTTOM-
C     MOST NA GRID POINTS AS PER AESD PE MODEL BY BROCK  -  NORDA
C     TECH NOTE 12  -  JAN 78
C          (1)  ATTENUATION  MATRIX ATT IS CALCULATED IN SUBROUTINE
C               NEWMAT
C
      COMPLEX U(5000)
      DIMENSION ATT(5000)
C
      DO 10 I=1,NA
         U(IA+I) = U(IA+I) * ATT(I)
 10   CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE END (RA2)
C
C     THIS SUBROUTINE IS CALLED WHEN THE SOLUTION FIELD HAS REACHED
C     THE MAXIMUM RANGE SPECIFIED (RMAX).  THE SUBROUTINE SENDS
C     APPROPRIATE MESSAGES TO THE TERMINAL ANC STOPS EXECUTION.
C
      WRITE(6,899) RA2
      WRITE(6,900)
      WRITE(6,901)
C
      STOP
 899  FORMAT(/,' THE PROGRAM HAS REACHED RANGE R =',F8.1,' METERS.')
 900  FORMAT ( ///,
     *   5X,'END OF RUN',/,
     *  10X,'(1)  CUTPUT PRINTER FILE:',/,
     *  15X,'(A)  A FILE WITH FORMATTED OUTPUT HAS BEEN GENERATED.',/,
     *  15X,'(B)  THIS FILE IS READY TO BE SENT TO THE PRINTER.',/,
     *  15X,'(C)  THE FILENAME AND FILETYPE FOR THIS FILE ARE:',/,
     *  15X,'          IFDOUT        PRINTER',//,
     *  10X,'(2)  CUTPUT PLOTTER FILE:',/,
     *  15X,'(A)  A FILE WITH UNFORMATTED CUTPUT HAS BEEN GENERATED.',/,
     *  15X,'(B)  THIS FILE CONTAINS INPUT DATA FOR THE PLOT RCUTINE.',/,
     *  15X,'(C)  THE FILENAME AND FILETYPE FOR THIS FILE ARE:',/,
     *  15X,'          IFDOUT        PLOTTER',/,
     *  15X,'(C)  THE FILENAME AND FILETYPE FOR THE PLOT ROUTINE ARE:',/,
     *  15X,'          PLOTIFD       FORTRAN ')
 901  FORMAT (
     *  15X,'(E)  TO PLOT THE DATA, GO TO A GRAPHICS TERMINAL AND',/,
     *  15X,'     ENTER COMMANDS AS FOLLOWS:',/,
     *  15X,'            DEF STOR 1M     ',/,
     *  15X,'            I CMS           ',/,
     *  15X,'            FORTGI PLCTIFD  ',/,
     *  15X,'            PLOTIFD         ',/,
     *  15X,'     THEN FOLLOW PROGRAM PROMPTS.')
C
      RETURN
      END
```

APPENDIX B

RUNNING THE IMPLICIT FINITE-DIFFERENCE PROGRAM ON THE NPS COMPUTER

A. INTRODUCTION

This appendix describes one simple procedure for running the IFD program on the NPS computer.

B. COPYING FILES ONTO USER'S DISK

Four files are needed to run the IFD program; the filenames and filetypes are

IFD	FORTRAN
IFD	EXEC
PLOTIFD	FORTRAN
PLOTIFD	EXEC

The files are available from a computer account maintained by the underwater acoustics curriculum. To link with this account and obtain copies of the files the user should proceed as follows:

(1) Log on terminal

(2) Enter: CP LINK 0160P 191 195 RR

(3) When prompted for read password enter: UX

(4) Enter: ACC 195 C

(5) Enter: COPYIFD

At this point the four files should reside on the user's A disk.

C. RUNNING THE IFD PROGRAM

Before running the IFD program the user should define additional storage space, compile the program, and set up the input data file. To define additional storage space,

(1) Enter: DEF STOR 1M
(2) Enter: I CMS

These two commands need only be entered one time for each terminal session; the storage space will remain for the entire session.

To compile the program,

Enter: FORTGI IFD

The program need be compiled only one time unless the program is changed, in which case the new version should be recompiled.

The final step before running the program is setting up the input data file which has filename and filetype IFDIN DATAIN. The user must create or modify this file so that it contains input data as described in Section III.E.2 of this thesis. For more information concerning how to create or modify files see NPS Technical Note TN-VM-02 which is available in the computer consultant's office.

If the above steps are accomplished, the user can then run the program with

Enter: IFD

Shortly after entering this command the user will be prompted for a run identification. The run identification

is an arbitrary identification label that will appear on the output printer file. Enter run identification as desired.

At the end of the run the user is informed of the two output data files generated by the program. If desired by the user the output printer file (IFDOUT PRINTER) may be sent to the printer. The output plotter file, IFDOUT PLOTTER, serves as an input file for the plotting program.

D. RUNNING THE PLOTTING PROGRAM

A Tektronix-618 terminal is used to run the plotting program. The first step is to log onto the terminal in the normal manner and then define additional storage space by,

(1) Enter: DEF STOR 1M
(2) Enter: I CMS

The plotting program has filename and filetype PLOTIFD FORTRAN. To compile the program,

Enter: FORTGI PLOTIFD

Unless the program is changed it need only be compiled one time. To run the program

Enter: PLOTIFD

The user will be prompted for axes and smoothing information; enter values and responses as appropriate. The transmission loss curve will be displayed on the CRT screen, a hard copy may be obtained by pushing the HARD COPY button under the screen, and the screen may be cleared by pushing the ENTER key.

APPENDIX C: PLOTIFD PROGRAM LISTING

```
C
C      THIS PROGRAM PLOTS TRANSMISSION LOSS CURVES FOR THE IFD PROGRAM.
C         (1)  INPUT DATA FROM THE IFD PROGRAM IS READ FROM A FILE WITH
C              FILENAME AND FILETYPE:    IFDOUT  PLOTTER
C         (2)  TO RUN THIS PROGRAM GO TO A GRAPHICS TERMINAL AND ENTER
C              THE FOLLOWING COMMANDS:
C                         DEF STOR 1M
C                         I CMS
C                         FORTGI PLOTIFD     (IF NOT ALREADY COMPILED)
C                         PLOTIFD
C              THEN FOLLOW PROGRAM PROMPTS.
C
       COMPLEX U
       DIMENSION P(10000), R(10000)
C
       DATA NOU/52/, IY/'Y'/, L/0/
C
C      *** LABEL OF X-AXIS AT ORIGIN IS ZERO KILOMETERS
           X1 = 0.0
C      *** LABEL AT RIGHT OF X-AXIS IS RMAX
C          *** READ MAXIMUM RANGE
               READ(NOU,*) RMAX
           X3 = RMAX / 1000.0
C      *** GET INCREMENT OF X-AXIS IN KILOMETERS
           WRITE(6,900) X3
           READ(5,*) X2
C      *** GET LABEL AT TOP OF Y-AXIS IN DB
           WRITE(6,901)
           READ(5,*) YY3
           Y3 = -YY3
C      *** GET LABEL OF Y-AXIS AT ORIGIN IN DB
           WRITE(6,902)
           READ(5,*) YY1
           Y1 = -YY1
C      *** GET INCREMENT OF Y-AXIS IN DB
           WRITE(6,903) YY3, YY1
           READ(5,*) Y2
C      *** DOES USER WANT SMOOTHING?
           WRITE(6,904)
           READ(5,905) IANS
           IF ( IANS.EQ.IY ) GO TO 10
C      *** NO
           NPOINT = 0
           GO TO 20
C
C      *** HOW MANY POINTS USED TO SMOOTH?
 10        WRITE(6,906)
           READ(5,*) NPOINT
```

```
C
C 20  *** REAC SOLUTICN FIELD
          REAC(NOL,*,END=40) RA, ZR, U
          IF (RA.LE.0.0) GO TO 20
             L = L + 1
             P(L) = CABS(U)
             R(L) = RA
          GO TO 2C
C
C 40   CONTINUE
      IF ( NPCINT.EQ.0 ) GO TO 70
C
C     *** THIS SECTION SMOOTHS THE DATA
          L = L - NPOINT + 1
          DO 60 I=1,L
             FTEMF = 0.0
             RTEMF = 0.0
             DO 5C J=1,NPOINT
                PTEMP = PTEMF + P(I+J-1)
                RTEMP = RTEMP + R(I+J-1)
 50          CONTINUE
             P(I) = PTEMP / FLOAT(NPOINT)
             R(I) = RTEMP / FLOAT(NPOINT)
 60       CONTINUE
C
C     *** CALCULATE VALUES TO BE PLOTTED
C     ***             NOTE: THE TRANSMISSICN LOSS VALUES CALCULATED
C     ***             BELOW ARE THE INVERSE SIGN OF THE TRUE TL VALUES
C     ***             THIS IS DONE SO THAT HIGHER TL VALUES TEND
C     ***             TOWARDS THE MINUS Y DIRECTION.
C
 70   DO 80 I=1,L
         P(I) = 2C.0*ALOG10(P(I)) - 10.0*ALOG10(R(I))
         R(I) = R(I) / 1000.0
 80   CONTINUE
C
      CALL TEK618
      CALL NCBRDR
      CALL PAGE(14.0,10.0)
      CALL PHYSOR(3.5,2.9)
      CALL AREA2D(10.0,6.7)
      CALL HEIGHT(.25)
      CALL XTICKS(2)
      CALL YTICKS(2)
      CALL XNAME('RANGE (KM)$',100)
      CALL YNAME('PROPAGATION LOSS (DB)$',100)
C     CALL HEADIN ('THIS IS A HEADING$',100,1.2,2)
      CALL GRAF (X1,X2,X3,Y1,Y2,Y3)
      CALL CUFVE(F,P,L,0)
```

```
      CALL ENDPL(C)
      CALL DCNEPL
C
      STOP
C
 900  FORMAT(/,
     * ' THE X-AXIS ON THE FORTHCOMING TRANSMISSION LOSS CURVE RANGES',/
     *,' FROM RANGE  R = 0.0 KM  TO  RANGE R =',F5.1,' KM.',/
     * ' ENTER DESIRED X-AXIS INCREMENT:')
 901  FORMAT(/,' ENTER LOWEST DB LOSS VALUE ON Y-AXIS:')
 902  FORMAT(/,' ENTER HIGHEST DB LOSS VALUE CN Y-AXIS:')
 903  FORMAT(/,
     * ' THE Y-AXIS ON THE TL CURVE WILL RANGE FRCM A DB LOSS OF',F6.1,
     * ' DB',/,' TO',F6.1,' DB.',/,' ENTER DESIRED Y-AXIS INCREMENT:')
 904  FORMAT(/,' CO YOU WANT THE DATA SMOOTHEC?  ( ENTER: Y OR N )')
 905  FORMAT(A1)
 906  FORMAT(/,' HOW MANY POINTS DO YOU WANT AVERAGED AT EACH STEP TO ',
     * 'OBTAIN SMCOTHING?',/,' ( ENTER: 2 OR 3 OR ... )')
      END
```

APPENDIX D

EXAMPLE OF IMPLICIT FINITE-DIFFERENCE PRINTED OUTPUT

```
IFD PRINTED OUTPUT
RUN I.D. :  RUN 30, DEEP-TO-SHALLOW WATER, 8.5 DEG.

     GAUSSIAN STARTING FIELD
     (FOR FURTHER INFORMATION ON VARIABLES SEE MAIN IFD PROGRAM LISTING)
     FREQUENCY                            FRQ    =    25.00 HZ
     SOURCE DEPTH                         ZS     =    25.00 M
     RECEIVER DEPTH                       ZR     =    25.00 M
     DEPTH INCREMENT                      DZ     =     1.00 M
     REF SOUND SPEED                      CO     =  1500.00 M/S
     REF WAVE NUMBER                      XKO    =     0.10 1/M
     REF WAVELENGTH                       XLAMDA =    60.00 M
     MAX RANGE OF SOLUTION                RMAX   =  40000.0 M
     RANGE STEP ON HORIZONTAL SECTION     DRLVL  =    10.00 M
     MAXIMUM RANGE STEP                   DRMAX  =    60.00 M
     RECORDED RANGE STEP                  WDR    =   100.00 M
     DEPTH OF UPPER EDGE ART ATTENU LYR   ZABLYR =   750.00 M
     #VERTICAL POINTS IN GRID              N      =   1000


SOUND SPEED PROFILE IN WATER:
               DEPTH          SOUND SPEED

               0.0  M         1500.00 M/S
```

350.00 M 1500.00 M/S

LAYER	MAX DEPTH(M)	DENSITY(G/CM**3)	ATT(DB/WL)	SOUND SPEED
WATER	350.0	1.00	0.000004	SEE PROFILE ABOVE
SEDIMENT	1000.0	1.50	0.200000	1600.00

BOTTOM PROFILE:

RANGE	DEPTH
0.0 M	350.0 M
10000.0 M	350.0 M
12000.0 M	50.0 M
40000.0 M	50.0 M

RANGE = 10000.0 M

DEPTH	LOSS(DB)	U(I)	
50.00	81.70	(0.77277E-02	0.28103E-02)
100.00	86.33	(-0.48267E-02	-0.82565E-04)
150.00	82.62	(-0.71538E-02	-0.18895E-02)
200.00	77.90	(0.12576E-01	0.19843E-02)
250.00	85.01	(0.45182E-02	0.33360E-02)
300.00	67.84	(-0.40433E-01	-0.33553E-02)
350.00	66.93	(-0.44474E-01	-0.71366E-02)
400.00	77.05	(-0.13282E-01	-0.45756E-02)
450.00	86.86	(-0.34990E-02	-0.28944E-02)

500.00	97.76	(-0.39013E-03	-0.12333E-02)
550.00	112.59	(0.15877E-04	0.23406E-03)
600.00	97.93	(-0.65141E-03	0.10893E-02)
650.00	95.72	(-0.13758E-02	0.88724E-03)
700.00	97.76	(-0.12791E-02	-0.20006E-03)
750.00	98.72	(-0.31804E-03	-0.11149E-02)
800.00	96.40	(0.70890E-03	-0.13379E-02)
850.00	97.02	(0.12450E-02	-0.66062E-03)
900.00	102.41	(0.71570E-03	0.24985E-03)
950.00	111.83	(-0.99971E-05	0.25606E-03)
1000.00	999.90	(0.0	0.0)

RANGE = 20000.0 M

DEPTH	LOSS(DB)	U(I)	
50.00	94.53	(0.24636E-02	-0.98700E-03)
100.00	98.62	(0.16240E-02	-0.33103E-03)
150.00	102.38	(0.10743E-02	0.54643E-04)
200.00	105.98	(0.68925E-03	0.17218E-03)
250.00	110.57	(0.40369E-03	0.11233E-03)
300.00	115.44	(0.23810E-03	0.20837E-04)
350.00	119.51	(0.14732E-03	0.26327E-04)
400.00	119.06	(0.57790E-04	0.14656E-03)
450.00	114.69	(-0.66006E-04	0.25225E-03)
500.00	112.93	(-0.16415E-03	0.27389E-03)
550.00	113.16	(-0.21721E-03	0.22228E-03)
600.00	114.52	(-0.22098E-03	0.14764E-03)
650.00	114.76	(-0.24573E-03	0.80022E-04)
700.00	114.73	(-0.25931E-03	-0.48798E-05)
750.00	115.06	(-0.22766E-03	-0.10299E-03)
800.00	117.20	(-0.14920E-03	-0.12605E-03)
850.00	121.46	(-0.55672E-04	-0.10583E-03)
900.00	129.09	(0.27763E-05	-0.49602E-04)
950.00	138.62	(0.14142E-04	-0.86376E-05)
1000.00	999.90	(0.0	0.0)

RANGE = 30000.0 M

DEPTH	LOSS(DB)	U(I)	
50.00	112.30	(-0.37351E-03	-0.19232E-03)
100.00	116.30	(-0.22408E-03	-0.14161E-03)
150.00	121.09	(-0.12386E-03	-0.89338E-04)

200.00	125.18	(-0.73035E-04	-0.61371E-04)
250.00	129.25	(-0.43901E-04	-0.40519E-04)
300.00	134.72	(-0.13981E-04	-0.28585E-04)
350.00	141.12	(-0.10268E-04	-0.11249E-04)
400.00	142.77	(-0.46205E-05	-0.11708E-04)
450.00	153.55	(-0.31555E-05	-0.18155E-05)
500.00	149.81	(0.52767E-05	-0.18675E-05)
550.00	143.67	(0.93225E-05	0.64674E-05)
600.00	152.63	(0.32720E-05	0.23772E-05)
650.00	151.49	(0.61160E-06	0.45745E-05)
700.00	152.27	(0.35996E-05	0.22024E-05)
750.00	150.93	(0.46321E-05	-0.16669E-05)
800.00	151.47	(0.14144E-05	0.44012E-05)
850.00	156.86	(0.61950E-06	-0.24087E-05)
900.00	161.31	(0.10207E-05	0.10838E-05)
950.00	173.60	(-0.10455E-06	0.34623E-06)
1000.00	999.90	(0.0	0.0)

RANGE = 40000.0 M

DEPTH	LOSS(DB)	U(I)	
50.00	129.70	(0.17632E-04	0.63051E-04)
100.00	133.43	(0.10855E-04	0.41190E-04)
150.00	137.05	(0.32401E-05	0.27889E-04)
200.00	141.88	(-0.49503E-05	0.15321E-04)
250.00	147.95	(-0.28856E-06	0.80017E-05)
300.00	146.88	(-0.67590E-05	0.60281E-05)
350.00	146.19	(-0.21010E-05	0.95760E-05)
400.00	150.57	(-0.58119E-05	0.11465E-05)
450.00	152.26	(-0.42927E-05	-0.23082E-05)
500.00	152.20	(-0.47761E-05	0.11340E-05)
550.00	151.56	(0.29176E-05	0.44073E-05)
600.00	153.59	(-0.58954E-06	-0.41400E-05)
650.00	155.15	(-0.31022E-05	-0.16101E-05)
700.00	156.51	(0.27945E-05	-0.10597E-05)
750.00	150.12	(0.49631E-05	0.37752E-05)
800.00	162.05	(-0.11591E-05	-0.10743E-05)
850.00	165.21	(0.47118E-06	0.99169E-06)
900.00	173.61	(-0.41723E-06	-0.82557E-08)
950.00	219.85	(-0.11341E-08	0.16897E-08)
1000.00	999.90	(0.0	0.0)

LIST OF REFERENCES

Brock, H.K., "The AESD Parabolic Equation Model," NORDA Technical Note 12, January, 1978.

Gerald, C.F., Applied Numerical Analysis, pp. 128-133, Addison-Wesley, 1980.

Hardin, R.H. and Tappert, F.D., "Applications of the Split Step Fourier Method to the Numerical Solution on Nonlinear and Variable Coefficient Wave Equations," SIAM Review, v. 15, p. 423, 1973.

Jensen, F. and Krol, H., "The Use of the Parabolic Equation Method in Sound Propagation Modeling," SACLANTCEN Memorandum SM-72, August, 1975.

Jensen, F.B. and Kuperman, W.A., "Environmental Acoustical Modeling at SACLANTCEN," SACLANTCEN Report SR-34, November, 1979.

Lee, D. and Botseas, G., "IFD: An Implicit Finite-Difference Computer Model for Solving the Parabolic Equation," NUSC Technical Report 6659, May, 1982.

Lee, D., Botseas, G., and Papadakis, J.S., "Finite-difference solution to the parabolic wave equation," JASA, v. 70, pp. 795-800, September, 1981.

Lee, D. and Gilbert, K.E., "Recent Progress in Modeling Bottom-Interacting Sound Propagation with Parabolic Equations," OCEANS, pp. 172-177, September, 1982.

Lee, D. and McDaniel, S.T., "A finite-difference treatment of interface conditions for the parabolic wave equation: The irregular interface," JASA, v. 73, pp. 1441-1447, May, 1983.

Lee, D. and Papadakis, J.S., "Numerical Solutions of Underwater Acoustic Wave Propagation Problems," NUSC Technical Report 5929, February, 1979.

McDaniel, S.T. and Lee, D., "A finite-difference treatment of interface conditions for the parabolic equation: The horizontal interface," JASA, v. 71, pp. 855-858, April, 1982.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Commandant (G-PTE-1/TP41) U.S. Coast Guard Washington, D.C. 20593	2
4.	Commander (ecv) ATTN: LT Larry Jaeger Seventh Coast Guard District Federal Building 51 S.W. 1st Ave. Miami, Florida 33130	6
5.	Superintendent ATTN: Dr. A.B. Coppens, Code 61Cz Naval Postgraduate School Monterey, California 93943	5
6.	Superintendent ATTN: Dr. J.V. Sanders, Code 61Sd Naval Postgraduate School Monterey, California 93943	2
7.	Superintendent ATTN: Dr. S.L. Garrett, Code 61Gx Naval Postgraduate School Monterey, California 93943	1
8.	Superintendent ATTN: LCDR C.R. Dunlap, USN(Ret), Code 68Du Naval Postgraduate School Monterey, California 93943	1

9. Superintendent 1
ATTN: Dr. R. Mendez, Code 53Mu
Naval Postgraduate School
Monterey, California 93943

10. Superintendent 1
ATTN: LT J.L. Blanchard, USN
SMC 2621
Naval Postgraduate School
Monterey, California 93943

11. Commanding Officer 1
ATTN: Dr. D. Lee
Naval Underwater Systems Center
New London, Connecticut 06320

12. Chief of Naval Research 1
ATTN: Dr. M McKissick
800 N. Quincy St.
Arlington, Virginia 22217

13. Dr. J. Andrews 1
NSTL Station
Bay St. Louis, Mississippi 39529

14. Dr. F. Jensen 1
SACLANT ASW Research Center
Viale San Bartolomeo 400
I 19026 -- La Spezia, Italy

15. LCDR Park Sae Hun 1
AIRASRON 101
768-78 ROKN, Pohang
Seoul, Korea

206775

Thesis
J2382 Jaeger
c.1
A computer program for solving the parabolic equation using an Implicit Finite-Difference solution method and incorporating exact interface conditions.